ISSN 0542 9951

10
1995

NAJSTARSZE PISMO INFORMATYCZNE W POLSCE – UKAZUJE SIĘ OD 1965 ROKU

informatyka

miesięcznik profesjonalistów

POLITECHNIKA ŚLĄSKA
Biblioteka Główna
w Gliwicach
P 1877/95

Sezon komputerowy 95/96 rozpoczęty

SOFTARG '95

Tak więc – mamy już za sobą pierwszą wielką imprezę sezonu komputerowego 95/96 – katowicki Softarg. Niewątpliwie najważniejszym elementem tej imprezy jest konkurs na najlepszy polski program roku; przypomnijmy zatem dla porządku jego wyniki: pierwsze miejsce przypadło gliwickiej firmie **ASKOM** za system wizualizacji nadzoru komputerowego procesów przemysłowych o nazwie **ASIX**, dwa równorzędne drugie zdobyły gliwicka spółka **P.A. Nowa** za zintegrowany system informacji o zagospodarowaniu przestrzennym oraz wrocławska **Elba** za system bankowy **„Skarb"**, zaś trzy trzecie **Micomp** z Katowic za pakiet **„Dodatki mieszkaniowe"**, **Pablo** z Warszawy za system obsługi gabinetów medycznych **„Znachor"** (doceniając pomysłowość i pożyteczność tego programu, pozwalam sobie wyrazić wątpliwość czy tytuł jego jest najbardziej fortunny: który z lekarzy zechce pokazać pacjentowi na ekranie swego komputera to słowo, mające przecież charakter epitetu?) oraz działające w tandemie warszawskie firmy **Edbud** i **MTM Digital** za moduł **„Kosztorysowanie"**.

Jak z tego widać, nagrodzone zostały programy, czy może lepiej powiedzieć – informatyczne rozwiązania problemów, skali raczej dużej, przy czym dość specjalistyczne. Niewątpliwie napisane świetnie, z wielkim talentem programistycznym i absolutnym profesjonalizmem – ale takie, których prywatny indywidualny użytkownik komputera raczej nie będzie szukał w sklepie; zresztą nie będzie miał szansy ich tam znaleźć. Trochę szkoda, że tak się stało – tym bardziej, że bez trudu w dziedzinie programów czysto użytkowych można było znaleźć coś godnego uwagi.

Czego nie nagrodzono ...

Oto przykłady. Uwagę moją zwrócił niezwykle interesujący **MARS Professional w. 1.0** – system analizy, przetwarzania, rozpoznawania i pomiarów obrazu wideo i mikroskopowych, opracowany przez firmę **CRI JOLANTA**, Zakład Robotyki i Komputerów z Gliwic.

System ten może wydatnie pomóc badaczowi w analizie zjawiska korozji, analizie zdjęć rentgenowskich, badaniu roślin, badaniach z zakresu densytometrii, farmakologii, histopatologii itp. Jest więc programem o dość szerokiej skali zastosowań; postaramy się wkrótce opisać go nieco bardziej szczegółowo.

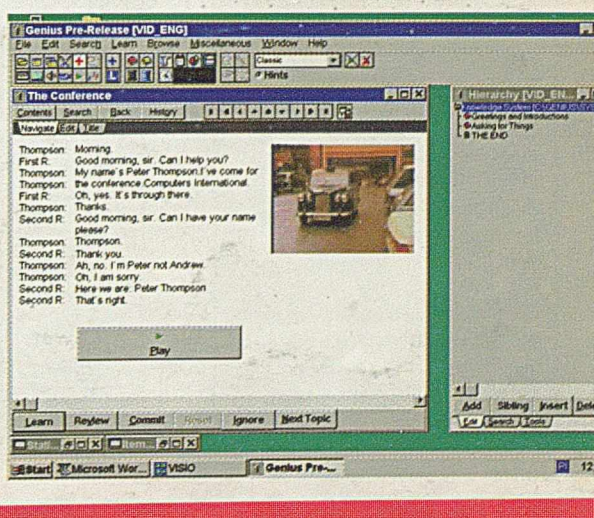
Drugi program, który wydaje mi się godny zainteresowania – to **Dyktando**, pochodzące z firmy **ZiP Soft** z Gliwic (do niedawna znanej głównie ze świetnych komputerowych słowników angielsko-polskich). Program ten wykorzystuje możliwości multimedialne komputera do bardzo atrakcyjnej nauki ortografii – od poziomu dla początkujących, do potwornie trudnych łamańców znanego ogólnopolskiego konkursu. Zdając sobie sprawę z faktu, że z pewnością dziś jeszcze w wielu szkołach nie ma odpowiednio silnych komputerów, pragnę jednak zwrócić uwagę na właściwy kierunek poszukiwań; warto też – jak sądzę – wyróżnić myślenie o tworzeniu produktu nie obliczonego przede wszystkim na szybki i duży zysk. Właściciel i główny programista firmy, Piotr Imiela, poczyna sobie w komputerowym biznesie wraz z małżonką, panią Zofią, wielce sprawnie i sympatycznie – dlatego też pozwalam sobie wymienić Go tu z nazwiska.

Akcent zagraniczny

Z łatwością wskazałbym co najmniej ze dwa jeszcze produkty dla indywidualnych użytkowników, ale zrobię to przy innej okazji – miejsca mało. Tym bardziej, że chciałbym jeszcze zwrócić uwagę Czytelników na dwa programy zagraniczne, które wydały mi się szczególnie ciekawe. Chodzi o spolszczoną wersję 2.0 izraelskiego wielojęzycznego procesora tekstu **Accent** oraz o przeniesiony wreszcie na platformę komputerów osobistych klasy PC słynny program DTP, **Calamus**.

Accent (rozpowszechniany przez firmę **Eureka** z Wrześni) jest programem rzeczywiście niesamowitym: jeśli ktoś używa w pracy kilku czy nawet kilkunastu języków, to jest to akurat dla niego. Jego najbardziej rozbudowana wersja daje użytkownikowi dostęp do ponad 30 języków, menu co najmniej w 19 językach, 150 krojów pism TrueType w 7 alfabetach, pomoc *on-line* w 11 językach, korektory ortografii dla 16 języków,

dokończenie na s. 32



Na Softargu '95 można było obejrzeć Geniusa – następcę słynnego programu edukacyjnego SuperMemo

WYDAWNICTWO SIGMA NOT

informatyka

nr 10 1995
Październik
Rok wydania XXXI

Miesięcznik
ISSN 0542-9951
INDEKS 36124

POLITECHNIKA ŚLĄSKA Biblioteka Główna w Gliwicach
P1877/95

REDAGUJE ZESPÓŁ:
mgr Bogdan **MIŚ**
(redaktor naczelny)
mgr Krystyna
KARWICKA-RYCHLEWICZ
(zastępca redaktora naczelnego)
mgr Teresa **JABŁOŃSKA**
(sekretarz redakcji)
Przemysław **BASZKIEWICZ**
(redaktor)
mgr Zdzisław **ŻURAKOWSKI**
(redaktor działu)
Alina **KLEPACZ** (sekretariat)

KOLEGIUM REDAKCYJNE:
prof. dr hab. Leonard BOLC,
mgr inż. Piotr FUGLEWICZ,
prof. dr hab. Jan GOLIŃSKI,
dr inż. Zenon KULPA,
doc. dr inż. Jan MULAWKA,
prof. dr hab. Wojciech OLEJNICZAK,
mgr inż. Jan RYŻKO,
dr Witold STANISZKIS,
dr inż. Jacek STOCHLAK,
prof. dr hab. Maciej STOLARSKI,
dr hab. Zdzisław SZYJEWSKI,
prof. dr hab. inż.
Ryszard TADEUSIEWICZ,
prof. dr hab. Jan WĘGLARZ

PRZEWODNICZĄCY
RADY PROGRAMOWEJ
prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004
Redakcja:
00-950 Warszawa,
ul. Ratuszowa 11, p. 628 i 644
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159

Materiałów nie zamówionych redakcja nie zwraca

Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach – w czystych kodach ASCII (Latin II, Mazovia) lub edytorach: Word 2, Word 6 – oraz odbitki na papierze.

Redakcja nie ingeruje w treść i formę ogłoszeń i innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

W sprawach ogłoszeń prosimy zwracać się bezpośrednio do Redakcji lub Działu Reklamy i Marketingu 00-950 Warszawa ul. Mazowiecka 12 telefon: 27-43-66 telefaks: 26-80-16

W numerze:

	Strona
Softarg '95	I okł.
publikacje	
Perspektywy obiektowych baz danych – *Waldemar Wieczerzycki, Robert Wrembel*	1
Perspektywy rozwoju kart płatniczych w Polsce (1). Typologia kart płatniczych – *Jacek Biskupski*	9
Typy z protekcją w języku Ada '95 – *Zbigniew Fryźlewicz*	14
Metody analizy i specyfikacji wymagań. Analiza i specyfikacja wymagań – *Krzysztof M. Sacha*	18
informacje	
INFO-FESTIWAL '95	26
Zaczęła się era Windows '95	28
AutoCAD rel. 13 po polsku **(kar)**	28
Apple nie boi się Windows '95 **(P.B.)**	29
CorelDRAW 6 zgodny z Windows '95 **(k)**	29
Quattro Pro 5.5 **(k)**	31
Zaprosili nas ...	32
Książki nadesłane	32

W najbliższych numerach:

- **Halina Kwaśnicka omawia książkę J. Chromiec i E. Strzemiennej „Sztuczna inteligencja. Metody konstrukcji i analizy systemów eksperckich".**
- **Jacek Biskupski kontynuuje cykl o perspektywach rozwoju kart płatniczych w Polsce i omawia problem zabezpieczenia systemów tych kart.**
- **Tomasz Pytlak opisuje podstawy składni języka CLIPS.**
- **Jan Kwiatkowski, Damian Misiak i Stanisław Szejko przedstawiają podejście strukturalne do analizy i projektowania systemów z uwzględnieniem rozszerzeń dla systemów czasu rzeczywistego.**

Warunki prenumeraty na 1996 r.

Zamówienia na prenumeratę czasopism wydawanych przez Wydawnictwo SIGMA-NOT można składać w dowolnym terminie. Mogą one obejmować dowolny okres czasu, tzn. dotyczyć dowolnej liczby kolejnych zeszytów każdego czasopisma.

Zamawiający może otrzymywać zaprenumerowany przez siebie tytuł począwszy od następnego miesiąca po dokonaniu wpłaty. Zamówienia na zeszyty sprzed daty otrzymania wpłaty będą realizowane w miarę możliwości – z posiadanych zapasów magazynowych.

Warunkiem przyjęcia i realizacji zamówienia jest otrzymanie z banku potwierdzenia dokonania wpłaty przez prenumeratora. Dokument wpłaty jest równoznaczny ze złożeniem zamówienia.

Wpłat na prenumeratę można dokonywać na ogólnie dostępnych blankietach w Urzędach Pocztowych (przekazy pieniężne) lub Bankach (polecenie przelewu), przekazując środki pod adres:
Wydawnictwo SIGMA-NOT Spółka z o.o.
Zakład Kolportażu
00-716 Warszawa, skr. poczt. 1004
konto:
PBK S.A. III 0/Warszawa nr 370015-1573-139-11

Wpłaty na prenumeratę przyjmują także wszystkie urzędy pocztowe nadawczo-odbiorcze oraz doręczyciele na terenie całego kraju

Na blankiecie wpłaty należy czytelnie podać nazwę zamawianego czasopisma, liczbę zamawianych egzemplarzy, okres prenumeraty oraz własny adres.
Na życzenie prenumeratora, zgłoszone np. telefonicznie, Zakład Kolportażu ul. Bartycka 20, 00-950 Warszawa, (telefony: 40-30-86, 40-35-89 oraz 40-00-21 wew. 249, 293, 299) wysyła specjalne blankiety zamówień wraz z aktualną listą tytułów i cennikiem czasopism.

Odbiorcy zagraniczni mogą otrzymywać czasopisma poprzez prenumeratę dewizową (wpłata dokonywana poza granicami Polski w dewizach, wg cennika dewizowego z cenami podanymi w dolarach amerykańskich) lub poprzez zamówioną w kraju prenumeratę ze zleceniem wysyłki za granicę (zamawiający podaje dokładny adres odbiorcy za granicą, dokonując równocześnie wpłaty w wysokości dwukrotnie wyższej niż cena normalnej prenumeraty krajowej).

Egzemplarze archiwalne (sprzedaż przelewowa lub za zaliczeniem pocztowym) można zamawiać pisemnie, kierując zamówienia pod adresem: Wydawnictwo SIGMA NOT, Spółka z o.o. Zakład Kolportażu, 00-716 Warszawa, ul. Bartycka 20, paw. B, tel. 40-37-31, natomiast za gotówkę można je nabyć w Klubie Prasy Technicznej w Warszawie ul. Mazowieckiej 12, tel. 26-80-17.

Istnieje możliwość zaprenumerowania 1 egz. czasopisma po cenie ulgowej przez indywidualnych członków stowarzyszeń naukowo-technicznych zrzeszonych w FSNT oraz przez uczniów zawodowych i studentów szkół wyższych. Blankiet wpłaty na prenumeratę ulgową musi być opatrzony na wszystkich odcinkach pieczęcią koła SNT lub szkoły.

W przypadku zmiany cen w okresie objętym prenumeratą Wydawnictwo zastrzega sobie prawo do wystąpienia o dopłatę różnicy cen oraz prawo do realizowania prenumeraty tylko w pełni opłaconej.
Cena jednego egzemplarza: normalna 5,00 zł (50 000 zł), ulgowa 3,75 zł (37 500 zł)
Wartość prenumeraty w zł:
Normalna: kwartalna 15,00 zł (150 000 zł), półroczna 30,00 zł (300 000 zł), roczna 60,00 zł (600 000 zł)
Ulgowa: kwartalna 11,25 zł (112 500 zł), półroczna 22,50 zł (225 000 zł), roczna 45,00 zł (450 000 zł)

Skład i druk: Drukarnia SIGMA NOT Sp. z o.o. z. 315/95

publikacje

Perspektywy obiektowych baz danych

Waldemar Wieczerzycki
Robert Wrembel
Francusko-Polska Wyższa Szkoła
Nowych Technik Informatyczno-Komunikacyjnych
Poznań

W najprostszym przypadku perspektywa obiektowa jest *klasą wirtualną,* to znaczy klasą, która nie ma własnych wystąpień (instancji). Klasa wirtualna jest zdefiniowana na podstawie innych klas, nazywanych *klasami bazowymi.* Instancje klas bazowych są dostępne przez perspektywę.

Specyfika perspektyw obiektowych

Perspektywy obiektowe, podobnie jak relacyjne, umożliwiają:

- autoryzację dostępu do obiektów;
- ułatwienie dostępu do obiektów, który w przypadku braku perspektyw jest realizowany za pomocą skomplikowanych zapytań;
- prezentację tych samych obiektów w różny sposób; dotyczy to zarówno stanów obiektów, reprezentowanych za pomocą wartości ich atrybutów, jak i zachowań obiektów, reprezentowanych za pomocą ich metod;
- logiczną niezależność obiektów i aplikacji, które odwołują się do nich przez perspektywy.

Specyfika perspektyw obiektowych polega na tym, że są one również wykorzystywane do restrukturyzacji obiektów i ewolucji schematu bazy danych, rozumianego jako zbiór klas i ich wzajemnych powiązań.

Restrukturyzacja obiektów jest możliwa dzięki temu, że perspektywa wywiedziona z określonej klasy bazowej może mieć inne atrybuty i metody niż klasa bazowa. W szczególnym przypadku perspektywa ma rozszerzony lub okrojony zbiór atrybutów i metod w stosunku do klasy bazowej.

Ewolucja schematu bazy danych jest możliwa przez tworzenie perspektyw, w których usunięto określone klasy, zmieniono ich definicje lub wzajemny porządek. Oczywiście, schemat bazowy pozostaje nie zmieniony. Część użytkowników realizuje dostęp do bazy danych przez własne schematy, definiowane przy użyciu perspektyw, podczas gdy pozostali – przez schemat bazowy.

W sierpniowym numerze INFORMATYKI zamieściliśmy artykuł Autorów pt. „Perspektywy relacyjnych baz danych"

Istotną korzyścią wynikającą ze stosowania perspektyw jest możliwość wstępnego przetestowania planowanych zmian schematu bazowego na odpowiednio skonstruowanych perspektywach.

Warto też podkreślić, że perspektywy ułatwiają korzystanie z heterogenicznych baz danych. W takich środowiskach mamy do czynienia z połączeniem różnych modeli danych. Perspektywy umożliwiają transformację jednego modelu w drugi i jednorodny dostęp do wystąpień klas w ogólności przechowywanych w różnych bazach danych.

Podstawowe pojęcia

Paradygmat obiektowy, stosowany w obiektowych bazach danych, opiera się na dobrze znanych pojęciach: klasy, wystąpienia i związku dziedziczenia. W celu przybliżenia perspektyw obiektowych bez danych jest konieczne wprowadzenie kilku dodatkowych pojęć, niezbędnych do analizy umiejscowienia klas i perspektyw we wspólnym grafie dziedziczenia [8].

Niech C oznacza klasę. Przez **typ** (C) oznaczamy tzw. *typ klasy,* będący zbiorem atrybutów i metod klasy wraz z ich dziedzinami. Dziedzinę atrybutu rozumiemy klasycznie, tj. jako zbiór wartości jakie może przyjmować, natomiast dziedzinę metody – jak zbiór wartości, które może wyznaczać.

Niech **prop** (C) oznacza zbiór atrybutów i metod klasy C.

Niech a oznacza atrybut lub metodę klasy. Przez $\mathbf{dom}_a$ (C) oznaczamy dziedzinę atrybutu lub metody a.

Niech o oznacza obiekt będący wystąpieniem klasy C. Przez **ext**(C) oznaczamy zbiór wszystkich obiektów będących wystąpieniami klasy C: **ext** $(C) := \{o \mid o \in C\}$.

Relacja podzbiorów

Dla dwóch klas $C1$ i $C2$, $C1$ jest podzbiorem klasy $C2$, co oznaczamy: $C1 \subseteq C2$, wtedy i tylko wtedy gdy:
ext $(C1) \subseteq$ **ext** $(C2)$ (lub $\Leftrightarrow (\forall o \in O)((o \in C1) \Rightarrow (o \in C2))$, gdzie O jest zbiorem wszystkich obiektów).

Relacja podtypów

Dla dwóch klas *C1* i *C2*, *C1* jest podtypem *C2* (a jednocześnie *C2* jest nadtypem *C1*, co oznaczamy: $C1 < C2$, wtedy i tylko wtedy gdy:
$(\mathbf{prop}\ (C1) \supseteq \mathbf{prop}\ (C2)) \wedge (\forall a \in \mathbf{prop}\ (C2))\ (\mathbf{dom}_a\ (C1) \subseteq \mathbf{dom}_a\ (C2))$.

Pierwszy warunek oznacza, po pierwsze, że podtyp musi mieć takie same atrybuty, jak nadtyp, a po drugie, że podtyp może mieć również dodatkowe atrybuty. Drugi warunek oznacza, że dziedziny atrybutów podtypu muszą być zawarte w dziedzinach atrybutów nadtypu.

Relacja podklas

Dla dwóch klas *C1*, *C2*, *C1* jest podklasą klasy *C2* (a jednocześnie *C2* jest nadklasą *C1*), co oznaczamy przez *C1* **jest** *C2*, wtedy i tylko wtedy gdy:
$(C1 < C2) \wedge (C1 \subseteq C2)$.

Pierwszy warunek oznacza, że atrybuty i metody podklasy muszą stanowić podtyp atrybutów i metod nadklasy. Drugi warunek oznacza, że każde wystąpienie podklasy jest jednocześnie wystąpieniem nadklasy.

Wykorzystując wprowadzone powyżej relacje, omówimy teraz podstawowe operatory obiektowe, które są wykorzystywane do definiowania klas wirtualnych.

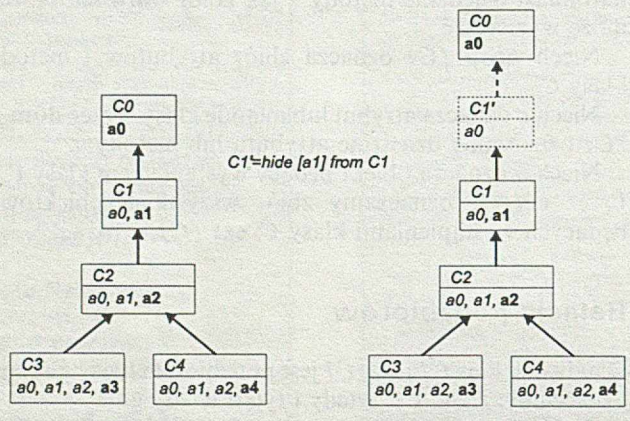
Operatory obiektowe

Operatory obiektowe można pogrupować w następujący sposób:

- operatory na typach, do których zaliczamy operator ukrycia i rozszerzenia,
- operatory na zbiorach, do których zaliczamy operator selekcji, unii, przecięcia i różnicy.

Operator ukrycia definiuje klasę wirtualną, która ma mniej atrybutów i (lub) metod niż klasa bazowa. Jego składnia jest następująca:
klasa_wirtualna = **hide** [*atrybut1, ..., metoda1, ...*] **from** *klasa_bazowa*;

Atrybuty wymienione po słowie kluczowym **hide** są niewidoczne w *klasie_wirtualnej*. W konsekwencji, między *klasą_bazową* a *klasą_wirtualną* zachodzą następujące zależności:
typ (*klasa_bazowa*) < **typ** (*klasa_wirtualna*) i
ext (*klasa_bazowa*) = **ext** (*klasa_wirtualna*).



Rys. 1. Klasa wirtualna zdefiniowana za pomocą operatora ukrycia

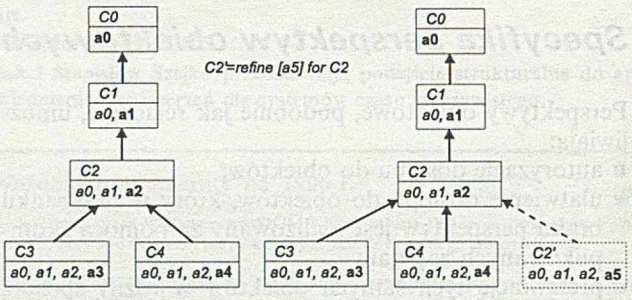
Oznacza to, że *klasa_wirtualna* jest nadtypem *klasy_bazowej* (ściśle mówiąc: typ klasy wirtualnej jest nadtypem typu klasy bazowej).

Jako przykład rozważmy hierarchię klas z rys. 1. Klasa wirtualna *C1'*, zdefiniowana za pomocą operatora ukrycia, jest nadtypem klasy bazowej *C1* i podtypem klasy *C0*. *a1, a2, a3, a4* oznaczają atrybuty klas (atrybuty pisane kursywą są dziedziczone).

Operator rozszerzenia definiuje klasę, która jest bardziej wyspecjalizowana niż klasa bazowa, tzn. że ma od niej więcej atrybutów lub(i) metod. Ponieważ klasa wirtualna nie ma własnych wystąpień, to dodawane atrybuty mogą być wyłącznie atrybutami wywiedzionymi. Oznacza to, że każdorazowe odwołanie się do nich wiąże się z koniecznością wyliczenia ich aktualnej wartości na podstawie wartości innych atrybutów i(lub) przy użyciu określonych metod. Operator rozszerzenia ma następującą składnię:
klasa_wirtualna = **refine** [*atrybut1, ..., metoda1, ...*] **for** *klasa_bazowa*;

Zgodnie z wprowadzonymi wcześniej relacjami, między *klasą_bazową* i *klasą_wirtualną* zachodzą następujące zależności:
typ (*klasa_wirtualna* < **typ** (*klasa_bazowa*) i
ext (*klasa_wirtualna*) = **ext** (*klasa_bazowa*).

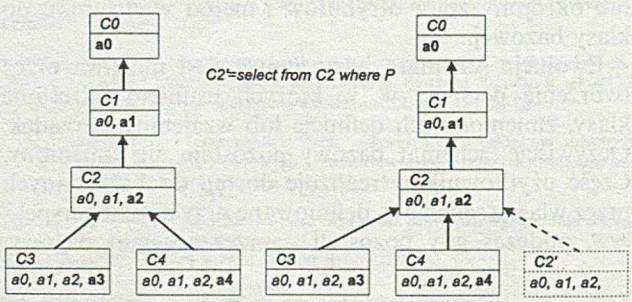
W konsekwencji, *klasa_wirtualna* jest podtypem *klasy_bazowej*. Zilustrowano to na rys. 2.



Rys. 2. Klasa wirtualna zdefiniowana za pomocą operatora rozszerzenia

Operator selekcji definiuje klasę wirtualną, przez którą są dostępne tylko te instancje klasy bazowej, które spełniają określony w nim warunek selekcji. Składnia operatora jest następująca:
klasa_wirtualna = **select from** *klasa_bazowa* **where** *warunek*;

Między *klasą_bazową* i *klasą_wirtualną* zachodzą następujące zależności:



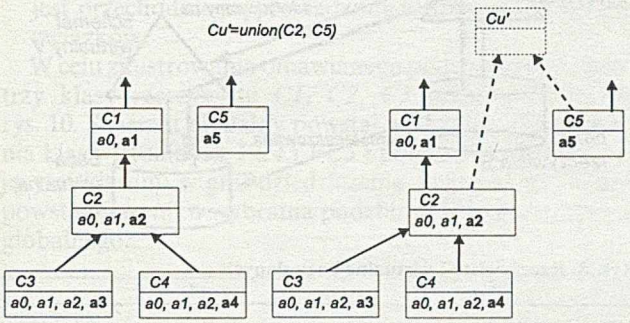
Rys. 3. Klasa wirtualna zdefiniowana za pomocą operatora selekcji

typ (*klasa_wirtualna*) = **typ** (*klasa_bazowa*) i
ext (*klasa_wirtualna*) ⊆ **ext** (*klasa_bazowa*).

Oznacza to, że *klasa_wirtualna* jest podtypem *klasy_bazowej*. Przykładowa klasa wirtualna zdefiniowana za pomocą operatora selekcji została przedstawiona na rys. 3.

Operator unii definiuje klasę wirtualną, przez którą są dostępne wystąpienia dwóch klas, sprowadzone do wspólnego typu. Operator ten ma następującą składnię:
klasa_wirtualna = **union** (*klasaA, klasaB*);

Typ klasy wirtualnej jest najmniejszym wspólnym nadtypem obu klas bazowych, tzn.
typ (*klasa_wirtualna*) = **typ** (*klasaA*) ∩ **typ** (*klasaB*),
natomiast zbiór wystąpień klasy wirtualnej zawiera:
ext (*klasa_wirtualna*) = **ext** (*klasaA*) ∪ **ext** (*klasaB*),
Oznacza to, że *klasa_wirtualna* jest nadtypem obu klas bazowych. Zostało to zilustrowane na rys. 4.

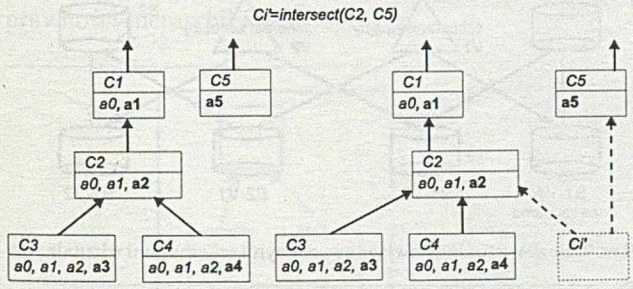


Rys. 4. Klasa wirtualna zdefiniowana za pomocą operatora unii

Operator przekroju definiuje klasę wirtualną, przez którą są widoczne te obiekty klas bazowych, które po sprowadzeniu do wspólnego typu mają takie same wartości w obu klasach. Operator ten ma następującą składnię:
klasa_wirtualna = **intersect** (*klasaA, klasaB*);

Typ klasy wirtualnej jest największym wspólnym podtypem, tzn.
typ (*klasa_wirtualna*) = **typ** (*klasaA*) ∪ **typ** (*klasaB*),
natomiast zbiór wystąpień klasy wirtualnej zawiera:
ext (*klasa_wirtualna*) = **ext** (*klasaA*) ∩ **ext** (*klasaB*),

Oznacza to, że *klasa_wirtualna* jest podtypem obu klas bazowych. Przykład umieszczenia klasy wirtualnej *Ci'* zdefiniowanej za pomocą operatora przekroju w grafie dziedziczenia został przedstawiony na rys. 5.

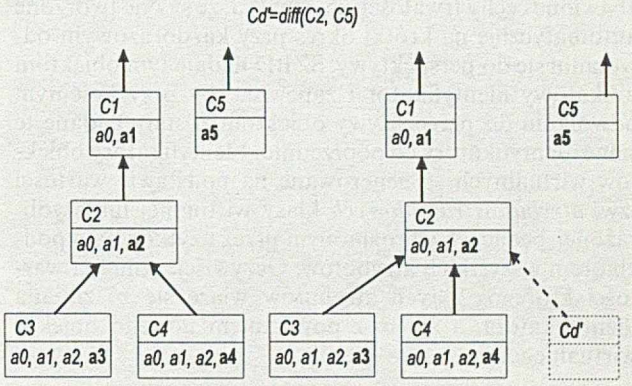


Rys. 5. Klasa wirtualna zdefiniowana za pomocą operatora przekroju

Operator różnicy definiuje perspektywę, przez którą są widoczne wszystkie wystąpienia klasy bazowej *A*, których wartości są różne od wartości wystąpień klasy bazowej *B*. Operator ten ma następującą składnię:
klasa_wirtualna = **minus** (*klasaA, klasaB*);

Typ klasy wirtualnej jest taki sam jak typ klasy bazowej *A*, tzn.
typ (*klasa_wirtualna*) = **typ** (*klasaA*),
natomiast zbiór wystąpień klasy wirtualnej zawiera:
ext (*klasa_wirtualna*) ⊆ **ext** (*klasaA*),

Oznacza to, że *klasa_wirtualna* jest podtypem *klasyA*. Przykład umieszczenia klasy wirtualnej *Cd'* zdefiniowanej za pomocą operatora różnicy w grafie dziedziczenia przedstawiono na rys. 6.



Rys. 6. Klasa wirtualna zdefiniowana za pomocą operatora różnicy

Podstawowe podejścia

Jak wspomniano, w najprostszym przypadku perspektywa jest pojedynczą klasą wirtualną zdefiniowaną przy użyciu przedstawionych powyżej operatorów [2, 11]. W ogólności, perspektywy mogą być konstruowane na bazie wielu klas, będących fragmentem schematu bazy danych, lub wręcz na całym schemacie. Spośród podejść zaproponowanych w literaturze warto wyróżnić następujące:

- perspektywę definiuje się jako schemat wirtualny będący zbiorem klas wirtualnych, połączonych związkiem dziedziczenia.
- perspektywę definiuje się jako schemat wirtualny złożony z klas rzeczywistych i wirtualnych, połączonych związkiem dziedziczenia,
- funkcje perspektywy uzyskuje się przez rozszerzenie paradygmatu obiektowego.

Omówimy teraz krótko specyfikę każdego z tych podejść.

Perspektywa jako pojedyncza klasa wirtualna

W podejściu tym, poza klasami wirtualnymi, wprowadza się tzw. *klasy wyobrażone*. Są one definiowane jako połączenie kilku klas bazowych, a zatem zawierają ich wszystkie atrybuty lub ich określone podzbiory. Klasy wyobrażone można w pewnym zakresie porównać do perspektyw relacyjnych definiowanych przy użyciu operatora połączenia.

Zarówno klasy wirtualne jak i wyobrażone mogą mieć dodatkowe atrybuty, które są nazywane *atrybutami wirtualnymi*. Wartości tych atrybutów nie są przechowywane

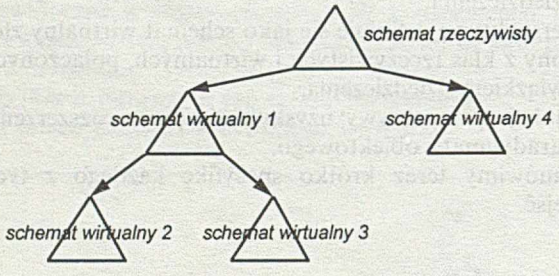
w bazie danych, lecz przy każdorazowym odwołaniu się do nich są one wyliczane na podstawie wartości atrybutów klas bazowych.

Zapytania kierowane do klas wirtualnych i wyobrażonych wyznaczają pojedyncze obiekty, zwane *obiektami wirtualnymi,* lub zbiory obiektów. Obiekty wirtualne zachowują się klasycznie – można zatem wysyłać do nich komunikaty uaktywniające odpowiednie metody. Obiekty wirtualne, w pewnych szczególnych sytuacjach mogą zostać *zmaterializowane* (ang. *materialized*) zwiększając w ten sposób efektywność pracy systemu.

Wystąpienia klas wirtualnych i wyobrażonych są pozbawione cechy trwałości, co oznacza, że są one tworzone automatycznie na krótki okres przy każdorazowym odwołaniu się do perspektywy. SZBD nadaje tym obiektom unikatowy identyfikator i zapewnia, by przy kolejnym odwołaniu do perspektywy obiektom zostały nadane te same identyfikatory co poprzednio. Identyfikatory obiektów wirtualnych są generowane na podstawie wartości tzw. *atrybutów rdzeniowych* klasy wirtualnej lub wyobrażonej, będących wyróżnionym przez użytkownika podzbiorem wszystkich atrybutów. Oczywiście, zmiana wartości któregoś z tych atrybutów wiąże się ze zmianą identyfikatora, a zatem z powstaniem nowego obiektu wirtualnego.

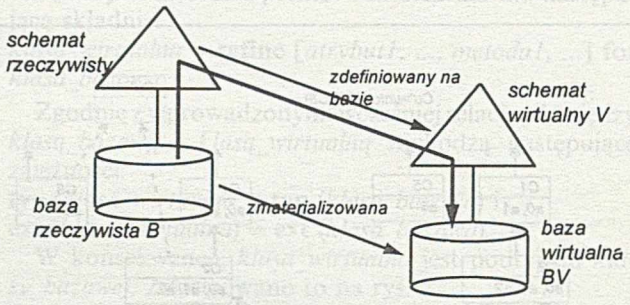
Perspektywa jako schemat wirtualny nie zawierający klas bazowych

W podejściu tym perspektywa w ogólności składa się z wielu klas wirtualnych i wyobrażonych, powiązanych wzajemnie związkami dziedziczenia i kompozycji, tworzących tzw. *schemat wirtualny* [9]. Schemat wirtualny jest wyprowadzany ze schematu bazowego, którym może być inny schemat wirtualny lub schemat rzeczywisty (tj. schemat złożony z klas rzeczywistych). Przykładowa hierarchia schematów wirtualnych została przedstawiona na rys. 7.



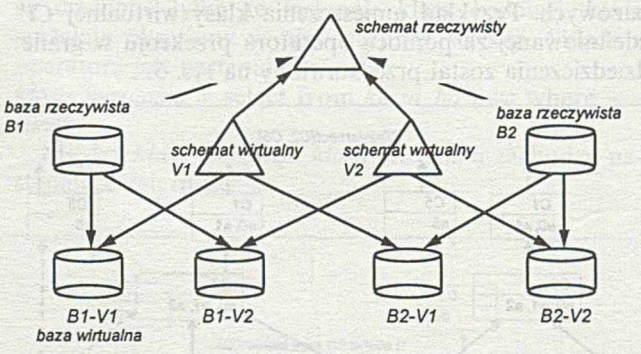
Rys. 7. Hierarchia schematów wirtualnych

Omawiane podejście jest implementowane w systemie *02*. Umożliwiono w nim jednoczesne istnienie wielu niezależnych baz danych, rozumianych jako zbiory trwałych obiektów, oraz wielu niezależnych schematów rzeczywistych, rozumianych jako definicje klas i powiązań między nimi. Powiązanie schematu z bazą danych jest realizowane dynamicznie: schemat wybiera się poleceniem **set schema** *<nazwa_schematu>*, natomiast odpowiadającą mu bazę – poleceniem **set base** *<nazwa_bazy>*. Dotyczy to również schematów wirtualnych, wywiedzionych bezpośrednio lub pośrednio z określonych schematów rzeczywistych. Wybranie schematu rzeczywistego wiąże się z przypisaniem mu odpowiedniej bazy rzeczywistej, natomiast wybranie schematu wirtualnego – bazy wirtualnej. Ze względów efektywnościowych, w przypadku wyboru bazy wirtualnej, *SZBD* automatycznie materializuje wszystkie zawarte w niej obiekty wirtualne. Zilustrowano to na rys. 8. Schemat wirtualny *V* wywiedziono ze schematu rzeczywistego. Schemat rzeczywisty jest wiązany z bazą rzeczywistą *B*. W przypadku wybrania schematu wirtualnego *V* jest materializowana baza wirtualna *BV*. Jej wystąpienia są wystąpieniami bazy rzeczywistej „widzianymi" przez schemat wirtualny *V*.



Rys. 8. Rzeczywiste i wirtualne bazy danych

Ponieważ, jak wspomniano, w systemie jednocześnie może istnieć wiele schematów i baz danych, w tym schematów i baz rzeczywistych, istnieje możliwość przeglądania różnych baz danych przez ten sam schemat wirtualny – i odwrotnie, tej samej bazy danych przez różne schematy. Zilustrowano to na rys. 9. Z jednym schematem rzeczywistym są związane dwie bazy rzeczywiste *B1* i *B2*. Ze schematu rzeczywistego wywiedziono dwa schematy wirtualne: *V1* i *V2*. Powiązanie baz *B1* i *B2* ze schematami *V1* i *V2* powoduje materializację czterech różnych baz wirtualnych.



Rys. 9. Bazy wirtualne powiązane z różnymi schematami wirtualnymi

Perspektywa jako schemat wirtualny zawierający klasy rzeczywiste i wirtualne

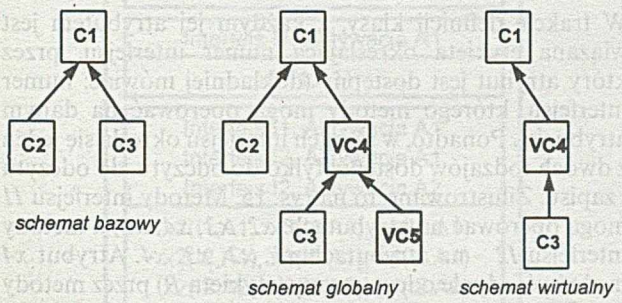
W tym podejściu perspektywa składa się z klas rzeczywistych i wirtualnych, które są umieszczone w odpowiadają-

cym jej pojedynczym schemacie wirtualnym. Zbiór wszystkich klas rzeczywistych wraz ze związkami dziedziczenia i kompozycji jest nazywany schematem rzeczywistym, natomiast zbiór wszystkich klas rzeczywistych i wirtualnych wraz ze związkami dziedziczenia i kompozycji – schematem globalnym. Podobnie jak w poprzednim podejściu, klasą bazową dla klasy wirtualnej może być klasa rzeczywista lub inna klasa wirtualna.

Perspektywa (schemat wirtualny) jest podschematem schematu globalnego. Charakteryzuje się następującymi cechami:

- ma unikatowy identyfikator,
- klasy perspektywy są również klasami schematu globalnego,
- w perspektywie są zawarte tylko te związki typu *jest* między klasami, które są bezpośrednio wyprowadzalne ze schematu globalnego; np. związek *jest* między klasami $<C1, C2>$ istnieje w perspektywie tylko wtedy, gdy istnieje również w schemacie globalnym lub jest przechodnio wyprowadzony z istniejących w nim związków.

W celu zilustrowania omawianego podejścia rozważmy trzy klasy rzeczywiste *C1, C2, C3* przedstawione na rys. 10. Schemat globalny powstał w wyniku zdefiniowania klas wirtualnych *VC4* i *VC5* i połączenia ich z istniejącymi klasami w graf dziedziczenia. Schemat wirtualny powstał w wyniku wybrania podzbioru klas ze schematu globalnego.



Rys. 10. Schematy: rzeczywisty, globalny i wirtualny

Każda perspektywa musi zapewniać spójność związków: dziedziczenia i kompozycji w stosunku do schematu globalnego. Spójność ta jest weryfikowana dwoma ortogonalnymi względem siebie kryteriami: kryterium poprawności hierarchii dziedziczenia oraz kryterium poprawności hierarchii kompozycji [8].

Kryterium poprawności grafu dziedziczenia

W tym kryterium są wykorzystywane trzy szczególne cechy związku dziedziczenia: nadmiarowość, obowiązkowość, zgodność. Związek jest *nadmiarowy* (ang. *redundant*), jeżeli jego usunięcie nie zmienia schematu powiązanych nim klas. Związek jest *obowiązkowy* (ang. *mandatory*), jeżeli jego usunięcie zmienia schemat klas w stosunku do schematu globalnego. Związek jest *niezgodny* (ang. *incompatible*), jeżeli nie istnieje bezpośrednio lub pośrednio pomiędzy odpowiednimi klasami schematu globalnego. Zilustrowano to na rys. 11. Schemat globalny zawiera pięć klas (rys. 11a). Perspektywa z rys. 11b zawiera związek *jest* między klasą *C4* i *C1*, który można usunąć, nie tracąc informacji o tym, że klasa *C4* jest podklasą *C1*. Perspektywa z rys. 11c jest niekompletna, ponieważ brakuje w niej związku *jest* między klasą *C4* i *C5*. W efekcie utracono więc informację o tym, że klasa *C4* dziedziczy jednocześnie z *C3* i *C5*. Rys. 11d przedstawia niezgodny schemat, ponieważ związek *jest* między klasą *C3* i *C5* nie istnieje w schemacie globalnym.

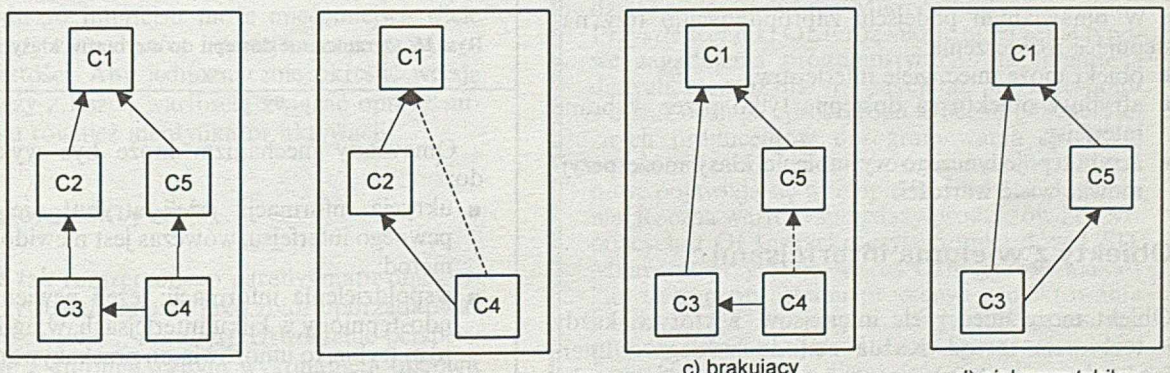
Perspektywa jest poprawna w płaszczyźnie dziedziczenia, jeśli:

- nie zawiera związków nadmiarowych,
- zawiera wszystkie obowiązkowe,
- nie zawiera związków niezgodnych.

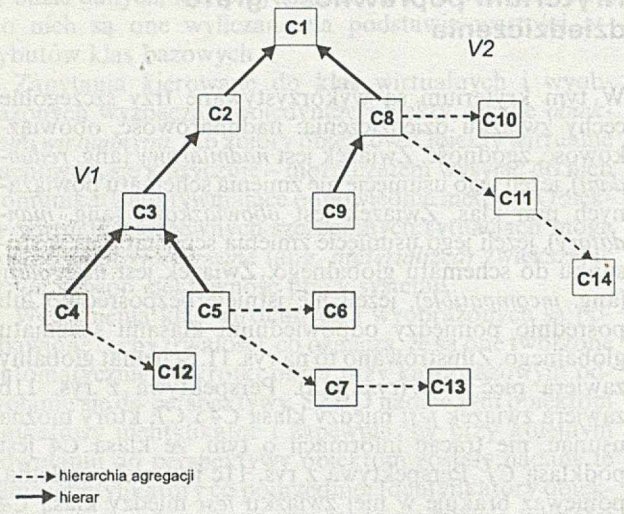
Kryterium poprawności hierarchii kompozycji

W celu wprowadzenia kryterium poprawności hierarchii kompozycji jest konieczne zdefiniowanie tzw. *domknięcia perspektywy*. Nieformalnie powiemy, że *perspektywa jest domknięta*, jeśli zawiera wszystkie klasy, będące dziedzinami wszystkich atrybutów klas zawartych w perspektywie. Dotyczy to przechodnio również dziedzin atrybutów klas, które same występują jako dziedziny innych atrybutów.

Perspektywa spełnia kryterium poprawności hierarchii kompozycji, jeżeli jest domknięta. Jako przykład rozważmy perspektywy *V1* i *V2* z rys. 12. Perspektywa *V1* jest domknięta, ponieważ dziedziny atrybutów klas *C4* i *C5*, tj. klasy *C6, C7, C12, C13,* również należą do tej perspektywy. Perspektywa *V2* nie jest domknięta, ponieważ klasa *C11*, która jest dziedziną atrybutu klasy *C8*, nie należy do perspektywy (podobnie, do perspektywy nie należy klasa *C14*, będąca dziedziną atrybutu klasy *C11*).



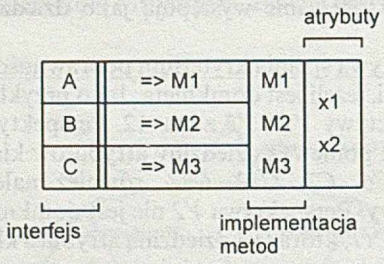
Rys. 11. Cechy związku dziedziczenia



Rys. 12. Kryterium poprawności hierarchii kompozycji

Rozszerzenia paradygmatu obiektowego

Ostatnie z omawianych podejść istotnie różni się od trzech poprzednich i wymaga rozszerzenia paradygmatu obiektowego [12]. W klasycznym paradygmacie obiektowym interfejs obiektu zawiera metody, z których każda ma swoją nazwę, sygnaturę i implementację. Zilustrowano to na rys. 13. Interfejs przedstawionej na nim klasy zawiera trzy metody: *A, B* i *C*. W klasie zawarto trzy implementacje metod: *M1, M2* i *M3*, związane odpowiednio z metodami *A, B* i *C*, co oznaczono symbolem ⇒. Klasa ma dwa atrybuty: *x1* i *x2*.



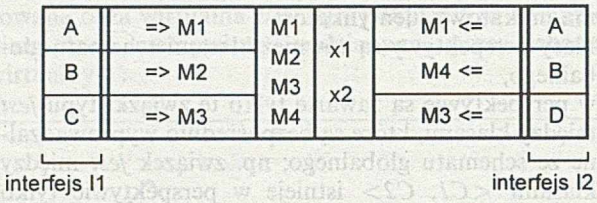
Rys. 13. Klasa w podejściu klasycznym

W omawianym podejściu zaproponowano trzy następujące rozszerzenia:

- obiekt może mieć wiele interfejsów,
- atrybuty obiektu są dostępne tylko przez wybrane interfejsy,
- atrybut pojedynczego wystąpienia klasy może przyjmować wiele wartości.

Obiekty z wieloma interfejsami

Obiekt może mieć wiele interfejsów, z których każdy jest zbiorem metod. Każda metoda każdego z interfejsów jest niezależnie związana z odpowiednią implementacją. Klasa z rys. 13 wyposażona w dwa interfejsy została przedstawiona na rys. 14. Każdy z nich składa się z trzech metod: *A* jest implementowana jednakowo w obu interfejsach przez *M1*, metoda *B* w pierwszym interfejsie jest implementowana przez *M2*, natomiast w drugim – przez *M4*. Każdy z interfejsów ma ponadto specyficzną dla siebie metodę, która nie istnieje w innym interfejsie (metoda *C* z pierwszego interfejsu i metoda *D* z drugiego interfejsu), ale obie mają tę samą implementację *M3*.

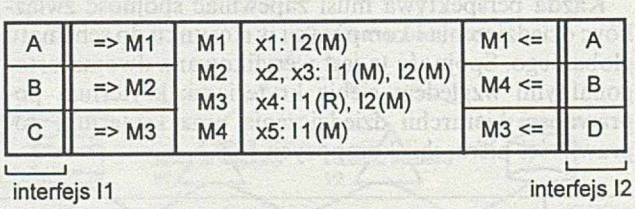


Rys. 14. Klasa z wieloma interfejsami

Wprowadzenie w definicji klasy wielu interfejsów powoduje, że jej wystąpienia mogą się zachowywać różnie w różnych kontekstach, w zależności od stosowanego interfejsu.

Ograniczanie dostępności atrybutów obiektów przez interfejs

W trakcie definicji klasy, z każdym jej atrybutem jest wiązana etykieta określająca numer interfejsu, przez który atrybut jest dostępny (dokładniej mówiąc: numer interfejsu, którego metody mogą operować na danym atrybucie). Ponadto, w ramach interfejsu określa się jeden z dwóch rodzajów dostępu: tylko do odczytu, do odczytu i zapisu. Zilustrowano to na rys. 15. Metody interfejsu *I1* mogą operować na atrybutach: *x2, x3, x4, x5*, a metody interfejsu *I2* – na atrybutach *x1, x2, x3, x4*. Atrybut *x4* może być jedynie odczytywany (etykieta *R*) przez metody interfejsu *I1*. Ograniczenie to nie obowiązuje w przypadku interfejsu *I2* (etykieta *M*).



Rys. 15. Ograniczenie dostępu do atrybutów klasy

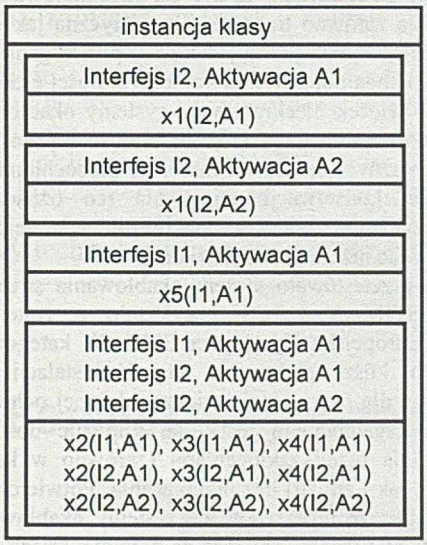
Omawiany mechanizm może być wykorzystywany do:

- ukrycia informacji; jeżeli atrybut nie ma etykiety pewnego interfejsu, wówczas jest niewidoczny dla jego metod;
- współdzielenia informacji; jeżeli pewien atrybut jest udostępniony w kilku interfejsach, w ogólności w różnym trybie, to umożliwia on przekazywanie informacji między zbiorami metod zawartych w różnych interfejsach.

Wersje danych

Istotą tego rozszerzenia jest możliwość modelowania wielu wartości pojedynczego atrybutu obiektu nazywanych *wersjami atrybutu* (ang. *attribute version*), z których każda odpowiada innej aktywacji tego samego interfejsu. Aktywacja interfejsu jest szczególną operacją, która powoduje utworzenie nowego zestawu wersji wszystkich atrybutów obiektu, wykorzystywanych przez metody zawarte we wskazanym parametrem aktywacji interfejsie. Wersjowanie atrybutów jest zatem dwupoziomowe. Po pierwsze, różnym interfejsom odpowiadają różne wersje atrybutów. Po drugie, w ramach określonego interfejsu istnieje możliwość operowania na różnych zestawach wersji atrybutów, wskazując identyfikator jednej z poprzednio wykonanych aktywacji tego interfejsu.

Wartość atrybutu jest jednoznacznie wyznaczona przez parę (*interfejs I, aktywacja A*) lub, innymi słowy, wartość atrybutu *val (attr)* jest funkcją interfejsu *I* i aktywacji *A*, tj. *val (attr) = f (I, A)*.



Rys. 16. Wielowersyjne atrybuty

W celu zilustrowania wersji atrybutów rozważmy klasę z rys. 15, która ma dwa interfejsy: *I1* i *I2*. Przyjmijmy, że zrealizowano dwie aktywacje interfejsu *I2* i jedną aktywację interfejsu *I1*. Pojedyncze wystąpienie klasy przedstawiono na rys. 16. Każdy z interfejsów wyznacza inną wartość zestawu atrybutów *x1, x2, x3, x4*. Ponieważ w ramach jednego interfejsu może mieć miejsce wiele aktywacji, każdy z atrybutów interfejsu może przyjmować wiele wartości. Aby jednoznacznie określić wersję atrybutu, należy z każdą wartością związać oprócz numeru interfejsu również identyfikator aktywacji.

Definicja perspektywy

Na podstawie tak rozszerzonego paradygmatu obiektowego perspektywę definiuje się jako zbiór uporządkowanych par postaci: (*klasa, interfejs*). Utworzenie perspektywy wiąże się z automatycznym wykonaniem operacji aktywacji w odpowiednich klasach i w ramach odpowiednich interfejsów. Przykładowo, wystąpienia klasy *C* zawartej w dwóch perspektywach z interfejsem *I1* i w trzech perspektywach z interfejsem *I2* będą zawierały pięć wersji atrybutów dostępnych przez oba interfejsy, dwie wersje atrybutów dostępnych tylko przez interfejs *I1* i trzy wersje atrybutów dostępnych tylko przez interfejs *I2*.

LITERATURA

[1] Abiteboul S., Bonner A.: Object and Views; w: Proc. ACM SIGMOD Conference on Management of Data, 1991
[2] Bertino E.: A View Mechanism for Object-Oriented Databases; w: Advances in Database Technology – EDBT '92. 3rd International Conference on Extending Database Technology Proceedings
[3] Cood E.F.: The Relational Model for Database Management, Version 2. Addison-Wesley Publishing Company, Inc., 1990
[4] Date C.J.: An Introduction to Database Systems. Vol. 1, Addison-Wesley Publishing Company, Inc., 1990
[5] Khoshafian S., Chan A., Wong A., Wong H.K.T.: A Guide to Developing Client/Server SQL Applications. Morgan Kaufmann Publishers
[6] O2Views User Manual, v. 2, 1995
[7] Oracle7 Server SQL Language Reference Manual, 1992
[8] Rundensteiner E.: MultiView: A Methodology for Supporting Multiple View Schemata in Object-Oriented Databases, Tech. Rep. 92-07
[9] Santos C., Delobel C., Abiteboul S.: Virtual Schemas and Bases; w: International Conference on Extending Data Base Technology, 1994
[10] Santos C., Delobel C., Abiteboul S.: Design and Implementation of na Object-Oriented View Mechanism; w: Technical Report, INRIA, Rocquencourt, 1994
[11] Scholl M.H., Laasch C., Tresch M.: Updatable Views in Object-Oriented Databases, w: Proc. DOOD, 1991
[12] Shilling J.J., Sweeney P.F.: Three Steps to Views: Extending the Object-Oriented Paradigm; w: Proceedings of Object-Oriented Programming, Languages, Systems and Applications Conference (OOPSLA), 1989

Microsoft zapowiada

Dalszy rozwój OLE

- **To był dobry pomysł!**
- **580 produktów wykorzystuje dziś tę technikę**

Wprowadzona w 1991 r. technika OLE – zainstalowana już na ponad 35 milionach komputerów architektura oprogramowania modułowego – będzie przez Microsoft wciąż ulepszana i rozwijana – oświadczył David Vashkevitch (jakoś dziwnie swojsko brzmi to nazwisko...), główny architekt działu programowania Microsoftu.
Przyszłe techniki OLE rozwiązać mają takie typowe zagadnienia programistyczne, jak integracja danych, zarządzanie transakcjami, praca zespołowa oraz obsługa uproszczona. Obecnie 348 niezależnych producentów oprogramowania sprzedaje ponad 580 produktów wykorzystujących OLE, zaś nowe pojawiają się w tempie średnio dwóch dziennie. Roczna wartość sprzedaży produktów korzystających z OLE przekroczyła 6 miliardów USD. Microsoft przypisuje sukces OLE przede wszystkim swemu otwartemu procesowi projektowania (*Open Design Process*), w którym od wczesnych stadiów uczestniczą producenci oprogramowania, firmy korzystające z niego oraz organizacje przemysłowe. (k)

Ekranować czy nie?
Dylemat użytkownika

Kompatybilność elektromagnetyczna (EMC) jest zazwyczaj definiowana jako zdolność systemów i urządzeń elektronicznych do poprawnej pracy bez zakłócania innych urządzeń i systemów, z dostateczną odpornością na takie zakłócenia. Tak więc mamy tutaj dwa aspekty: emisję zakłóceń i odporność na zakłócenia.

Implementacja Europejskiej Dyrektywy EMC wywołała wiele dyskusji i nieporozumień. Początkowo dyskutanci skoncentrowali się na wadach i zaletach ekranowanych i nieekranowanych systemów okablowania strukturalnego.

Kompatybilność elektromagnetyczna może być osiągnięta na kilka sposobów:

- transmisja symetryczna i filtrowanie
- ekranowanie.

Celem transmisji symetrycznej jest przesyłanie równych co do wartości i przeciwnych co do znaku sygnałów w liniach nieekranowanej pary. Sygnały te z kolei indukują równe co do wartości i przeciwnie skierowane pola elektromagnetyczne. Pola te znoszą się wzajemnie i rezultatem jest brak jakiejkolwiek emisji z takiej nie ekranowanej pary. W przypadku zakłóceń sygnał indukowany w liniach pary jest równy co do wartości i co do znaku, taki sygnał nazywany jest sygnałem wspólnym. W układach wejściowych odbiornika następuje odrzucenie sygnału wspólnego zwane filtrowaniem, tak więc do odbiornika nie docierają żadne zakłócenia. Reasumując, im lepsza symetryzacja toru tym większa odporność na zakłócenia, tym mniejsza emisja zakłóceń.

Ekranowanie to zastosowanie klatki Faradaya, to znaczy ekran jest umiejscowiony wokół kabli i całego osprzętu pomocniczego (krosownice, przełącznice, wtyczki, gniazdka i inne). Dwa podstawowe problemy takiego rozwiązania to zapewnienie „szczelności „ ekranu i sposób jego uziemienia. Ekran jak by nie patrzeć, jest przewodnikiem i jeżeli nie jest prawidłowo zainstalowany może łatwo transmitować niepożądane sygnały, przewodzić lub emitować zakłócenia.

Prawidłowe ekranowanie to:

◊ Pełne ekranowanie złącz z pełnym ekranowaniem kabli na złączach, niedopuszczalne są żadne nieekranowane końcówki kabli. Dotyczy to również gniazd, przełącznic i krosownic.

◊ Bardzo dokładna instalacja kabli. Kable ekranowane folią są mniej wygodne niż kable z oplotem stąd trzeba bardzo wiele uwagi poświęcić dokładności prac instalacyjnych, tak aby nie przekroczyć minimalnego dopuszczalnego promienia zgięcia kabla, czy maksymalnej siły wciągania. Nieprzestrzeganie tych obostrzeń może spowodować przerwanie ekranu.

◊ Uziemiane. Dla częstotliwości powyżej 1 MHz jest wymagane, aby ekran był uziemiany co 1/10 długości fali. (długość fali o częstotliwości 1 MHz w kablu ekranowanym wynosi 21 metrów !!). Kolejnym ograniczeniem jest długość kabla uziemiającego, tak aby uniknąć wtrącenia wysokiej impedancji (przy częstotliwości 35 MHz ta długość wynosi 35 cm).

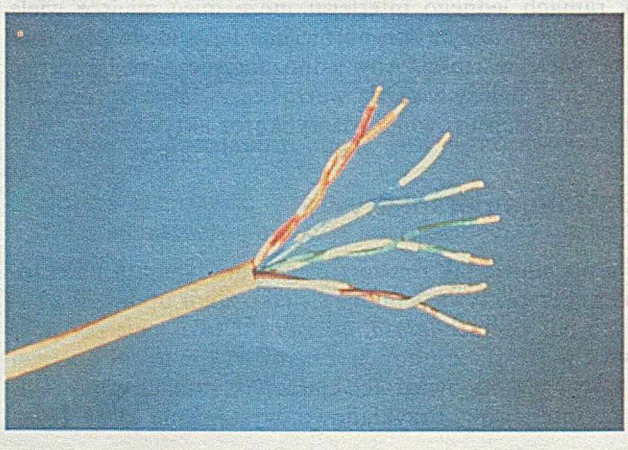
◊ Dostępność dobrego uziemienia i wolnego od zakłóceń źródła zasilania. Niskiej jakości uziemienie może być przyczyną emisji fal elektromagnetycznych i zakłóceń. **Ekranowanie nie może być lepszej jakości niż system uziemienia, do którego zostało podłączone. Jeżeli uziemienie jest zaszumione, to cały ekran będzie emitował szumy.** Zachowanie dobrych parametrów uziemienia w długim okresie czasu jest bardzo ważne i musi być często sprawdzane. Systemy uziemienia muszą być często sprawdzane i dla uniknięcia skutków korozji, wibracji i zmian temperaturowych wymagają stałej konserwacji. Często zdarza się że, uziemienie mimo nieprawidłowej instalacji daje znakomite efekty, ale jest źródłem tajemniczych kłopotów w przyszłości.

Systemy okablowania oparte na nieekranowanej skrętce wykorzystują zarówno transmisję symetryczną jak i filtrowanie. Transmisja symetryczna jest często postrzegana jako ekonomiczna metoda poprawienia kompatybilności elektromagnetycznej. Co więcej, nieekranowane systemy okablowania oferują dodatkowe zalety takie jak: łatwość instalacji (niższe koszty robocizny z uwagi na mniejszą pracochłonność) brak konieczności konserwacji uziemienia (co oznacza niższe koszty utrzymania), większe upakowanie - mniejsze koryta kablowe a więc niższa cena materiałów.

AT&T przetestowało system okablowania strukturalnego AT&T SYSTIMAX ® SCS i wykazało, że system ten jest zgodny z Europejską Dyrektywą EMC. W kategoriach ekonomicznych, koszt, praktyczna łatwość instalacji są realną przeciwwagą dla nadmiernej i nieuzasadnionej ochrony. Podstawowym zagadnieniem jest tutaj kupienie dostatecznego zabezpieczenia przed zakłóceniami (zarówno w kategoriach odporności, jak i emisji) bez przepłacania. Potwierdzeniem tej tezy jest fakt , że nieekranowane systemy okablowania mają większościowy udział w rynkach co najmniej trzech czwartych krajów europejskich. Dodatkowym atutem w przypadku międzynarodowych korporacji jest fakt, że narodowe normy uziemiania ekranu są różne w różnych krajach tak więc, nieekranowany system okablowania strukturalnego jest systemem prawdziwie otwartym i przenośnym.

Opracował: **Wojciech Apolinarski**

LITERATURA:
To shield or not to shield an end-users's dilemma ?? by Dr. T.C Tan, AT&T Bell Laboratories, AT&T Global Private Networks



0/25/95

Po latach niedostępności plastikowa płytka, nazywana przez nas zazwyczaj, choć nie zawsze słusznie, kartą kredytową, zaczyna być obecna również w Polsce. W konsekwencji tak długiej nieobecności, sprawne działanie systemów kart płatniczych polskich banków będzie wymagało od wydawców kart przede wszystkim ścisłej współpracy oraz zmian w bankowych systemach informatycznych i księgowych. Ponadto banki będą musiały ponieść ciężar szeroko zakrojonej promocji „pieniądza elektronicznego" w celu przekonania naszego – tradycyjnie „gotówkowego" – społeczeństwa do używania tej formy płatności ze wszystkimi jej blaskami i cieniami. Bliższa analiza zasad działania istniejących w świecie systemów kart płatniczych ujawnia luki w obowiązujących w naszym kraju przepisach prawnych, których uzupełnienie wydaje się niezbędne do właściwego ich funkcjonowania.

Czas, w jakim przyszło bankom polskim wdrażać swoje lokalne i międzynarodowe produkty kartowe, jest szczególnie trudny, gdyż zbiegł się z dokonującym się praktycznie w całym świecie przełomem w technologii kartowej. Wyraźny zmierzch technologii kart magnetycznych, zamortyzowanych moralnie po 25 latach użytkowania, nieusystematyzowany pakiet propozycji rozwiązań w technologii kart elektronicznych powodują uzasadnione rozterki strategów polskich systemów kartowych. Polityka największych organizacji kartowych na świecie jest już dziś jednoznacznie określona i z analizy publikowanych dokumentów wynika niezbicie, że cały świat zadecydował już o transformacji swoich systemów kartowych od kart magnetycznych do elektronicznych. Z drugiej strony zdani jesteśmy na stosowanie kart z paskiem magnetycznym jeszcze przez wiele lat – chociażby tylko dla realizowania płatności regulowanych kartami obywateli innych krajów świata.

Optymalne wydaje się wprowadzenie na naszym rynku rozwiązania tzw. karty hybrydowej, mającej jednocześnie pasek magnetyczny i układ elektroniczny dla kart o zasięgu międzynarodowym, oraz kart nastawionych zdecydowanie na obsługę elektroniczną w systemie krajowym. Pytanie: jakie informacje zostaną zapisane na pasku magnetycznym, a jakie w układzie elektronicznym – jest zagadnieniem postawionym właściwie wobec współczesnych systemów kartowych, również w Polsce. Odpowiedź na nie zdefiniuje charakter i zakres potencjalnych produktów bankowych oraz kierunki inwestycji. Bardzo istotną kwestią jest uwzględnienie wszystkich możliwości kart elektronicznych, jako technologii przyszłości, oraz przeprowadzenie długofalowego rachunku ekonomicznego, pozwalającego na właściwe określenie relacji nakładów do dochodów systemu kartowego.

Cykl o kartach składa się z trzech artykułów, z których pierwszy proponuje podział i definicję różnych typów kart, drugi traktuje o zabezpieczeniach tradycyjnie działających kart płatniczych, a trzeci jest próbą analizy technologii kart elektronicznych (ang. *smart cards*) do **zastosowań systemów płatniczych.**

Perspektywy rozwoju kart płatniczych w Polsce (1)

Typologia kart płatniczych

Jacek Biskupski
Ekspert PKN ds. kart identyfikacyjnych
DIGITCARD-UNICARD
Kraków

ARTYKUŁ REKOMENDOWANY

Pierwsze karty płatnicze, wystawione przez instytucje finansowe, pojawiły się w USA w roku 1947. Początkowo były to karty *Flatbush National Bank* – o zasięgu lokalnym oraz *Franklin National Bank* – o zasięgu ponadstanowym. Pierwszą organizacją, która określiła jednoznaczne zasady działania kart płatniczych charakteryzujących się możliwością zaciągnięcia krótkotrwałego kredytu, był *Diners Club*. W 1950 r. zaproponował on współpracę hotelom i restauracjom w Nowym Jorku, adresując swoją kartę głównie do często podróżujących ludzi interesu. Tam też w 1958 r. zaczął działać największy system kart tego typu – zwanych kartami T&E (*Travel and Entertainment*) – *American Express*. Fakt zaproponowania kart tylko wybranym klientom pozwolił na wprowadzenie opłat za wydanie karty oraz systemu prowizji od akceptantów, dokonujących transakcji. Z drugiej strony, kontrola jakości udzielanych kredytów pozwoliła na gwarantowanie zapłaty akceptantom (sklepom i usługodawcom), co przyczyniło się do szybkiego spopularyzowania tego typu kart wśród odbiorców, do których została ona skierowana.

Rekomenduje:
Prof. dr hab. Andrzej Baborski – J.M. Rektor Akademii Ekonomicznej we Wrocławiu

Idea popularnej bankowej karty kredytowej o zasięgu międzynarodowym, o podobnych zasadach używania dla każdego posiadacza, wystawianej przez różne banki, została zaproponowana przez Bank of America w roku 1966. Produkt o nazwie *BankAmericard*, stworzył pierwowzór najpopularniejszych do dziś kart kredytowych VISA (od 1977 r.) W tym samym roku inna grupa banków amerykańskich zaproponowała identyczny produkt pod nazwą *Interbank*, później *Master Charge*. Ten ostatni działa do dziś jako drugi, w rankingu liczby wydanych kart, system kartowy na świecie – karty *Master Card*.

Identyfikacyjna karta plastikowa

Z punktu widzenia technologii informatycznej każda karta plastikowa jest traktowana jako karta identyfikacyjna. Identyfikacyjne karty płatnicze (a więc również karty VISA, MASTER CARD, AMEX itd.), to płytki z laminowanego PCV o standardowych wymiarach $54 \times 86 \times 0{,}76$ mm, przygotowane zgodnie z normami ISO 7810-1985, 7811 (1-5)-1985, 7813-1995.

Charakteryzują się one nadrukiem, związanym z bankiem-wydawcą oraz (o ile karta działa poza jedną organizacją bankową) mają emblemat systemu kartowego, w ramach którego zostały wydane (np. VISA INT). Karty takie zawierają specjalny pasek umieszczony na jednej ze stron, na którym jej posiadacz składa wzór swojego podpisu. Jest też na nich najczęściej naniesiony pasek magnetyczny o koercji 300 Oe.

Karty mają kilka różnych zabezpieczeń, spośród których najczęściej używanymi w systemach kartowych obrotu międzynarodowego są hologram, nadruk UV, mikroprint.

Przed wydaniem posiadaczowi konta są personalizowane przez wytłoczenie lub nadruk płaski oraz kodowanie paska magnetycznego. Nadawane są im w ten sposób cechy unikatowe, jednoznacznie wiążąc tym samym kartę z kontem lub(i) klientem banku. Ostatnim etapem personalizacji jest w większości kart podpis posiadacza, składany przez niego na pasku do podpisu przy odbiorze karty.

Ze względu na zastosowanie karty w systemie bankowym oraz jej charakter można wyróżnić następujące podstawowe rodzaje kart spotykanych w systemach płatniczych:

- karty współtworzące produkty bankowe,
- karty narzędzia systemów bankowych,
- karty wstępnie opłacone,
- inne karty w systemach bankowych.

Warto zauważyć, że zaproponowany podział jest próbą podziału „obiektywnego", obejmującego wszystkie karty związane z bankowością, nie zaś karty płatnicze konkretnego systemu kartowego. Stąd też nie odpowiada on podziałom na typy w rozumieniu np. *VISA INT*, czy *PAY NOW* i *PAY LATER* według MASTERCARD/EUROPAY.

Karty współtworzące produkty bankowe

W zależności od typu produktu bankowego, z jakim związane są karty (co wiąże się ze sposobem rozliczania dokonanej transakcji), możemy wyróżnić następujące typy kart:

- **debetowe,**
- **obciążeniowe,**
- **kredytowe.**

Karta debetowa jest to karta pozwalająca na dokonywanie płatności za zakupione towary i usługi z posiadanych na koncie bankowym środków, tzn. równocześnie z transakcją następuje obciążenie rachunku jej posiadacza. Karta taka jest najczęściej ważna w zasięgu lokalnym, zwykle krajowym, choć istnieją również karty o zasięgu międzynarodowym – np. VISA ELECTRON czy Master Card MAESTRO, i może służyć przez określony czas lub do chwili wykorzystania całego wkładu. Posiadaczowi w zasadzie nie wolno nadużywać karty prowadząc zakupy za większe ilości środków niż zdeponował (w niektórych systemach istnieje możliwość zaciągnięcia kredytu w rachunku bieżącym zwanego *overdraft*). Karta jest obsługiwana elektronicznie przez terminale EFTPOS lub tradycyjnie (dla obniżenia kosztów) przez odciśnięcie na rachunkach samokopiujących w kopiarce ręcznej, tzw. *imprinterze* z autoryzacją telefoniczną. W przypadku kart obsługiwanych elektronicznie przeprowadzenie transakcji jest znacznie szybsze niż w przypadku karty kredytowej, np. w brytyjskim systemie SWITCH 95% transakcji jest obsługiwane w czasie do 5 sekund. Inną atrakcją jest w ich przypadku fakt gwarantowania zapłaty dla (prawie) każdej transakcji. Karta debetowa stanowi atrakcyjną i bezpieczniejszą alternatywę dla czeków. Bywa też nazywana **elektronicznym czekiem**. Choć jest to wciąż rozwijający się instrument płatniczy, okazało się, że na rynku europejskim kart typu debetowego jest już cztery razy więcej niż kart kredytowych [3], a w USA szacuje się, iż liczba transakcji dokonywana tego typu kartami przewyższa już liczbę transakcji kartami kredytowymi [7].

W Polsce przykładem funkcjonującej karty debetowej jest wprowadzona w 1993 r. lokalna karta płatnicza Banku Depozytowo-Kredytowego w Lublinie w systemie zorganizowanym w oddziale Banku w Stalowej Woli [5]. Posiadacze kont osobistych, którym wydano taką kartę, opłacają nią należność za zakupione towary w wybranych przez bank sklepach i punktach usługowych lub pobierają gotówkę. Na podstawie sporządzonych podczas transakcji rachunków oraz podpisu klienta następuje rozliczenie przez bank, a więc obciążenie konta posiadacza karty oraz uznanie konta sklepu. Na podobnej zasadzie, choć bez możliwości wypłat gotówkowych, działa karta *Hermes* Powszechnego Banku Kredytowego (PBK).

Również karty *VISA CLASSIC*, wydawane osobom fizycznym przez polskie banki są z punktu widzenia produktów bankowych kartami typu debetowego. Jednak sposób działania tych kart różni się w znacznym stopniu od klasycznych kart debetowych takich systemów, jak SWITCH czy CB card.

Mgr inż. Jacek BISKUPSKI ukończył w 1993 r. Wydział Inżynierii Materiałowej i Ceramiki w Akademii Górniczo-Hutniczej w Krakowie. W trakcie indywidualnego toku studiów odbywał dodatkowe zajęcia na wydziałach Elektroniki i Informatyki.

Szczególne zainteresowania dotyczą różnych rozwiązań systemu plastikowych kart (okresowe bilety komunikacji miejskiej, karty debetowe, pierwszy polski system rozliczeń bezgotówkowych zrealizowany na kartach elektronicznych TALAR).

W 1994 r. został powołany przez Prezesa PKN na eksperta w zakresie kart identyfikacyjnych. W 1995 r. otworzył przewód doktorski na temat bankowych aplikacji kart elektronicznych.

Współpraca PBK SA, PKO SA i PKO BP z Europay Int. ma doprowadzić w najbliższym czasie do wprowadzenia na rynek polski szerokiej oferty sprawdzonych na rynku europejskim kart debetowych systemu Europay.

Karta obciążeniowa (ang. *charge card, T&E card*) to typ karty płatniczej pozwalającej na dokonywanie płatności oraz pobieranie gotówki, której posiadacz karty aktualnie na koncie nie ma. Karta ta daje więc możliwość zaciągania krótkoterminowego kredytu, który powinien zostać spłacony w ciągu określonego czasu (najczęściej miesiąca). Karty obciążeniowe służą najczęściej regulowaniu kosztów podróży oraz delegacji służbowych, stąd ich określenie T&E card (*Travel and Entertainment*). Jest to produkt raczej ekskluzywny, ponieważ karta zawsze wydawana jest odpłatnie (opłata za kartę wynosi kilkaset USD rocznie), dodatkowo stosuje się opłaty za „przystąpienie do klubu użytkowników", a limit kredytowy jest ustalany indywidualnie. Sama karta poza prestiżem posiadania, dostarcza wielu dodatkowych przywilejów, takich jak bezpłatne ubezpieczenia w podróży, zniżki oraz pierwszeństwo w rezerwacjach hoteli, usługi medyczne czy pomoc prawną za granicą. W literaturze popularnonaukowej karty tego typu są często nieodróżniane od kart kredytowych, jednakże zarówno z punktu widzenia banku, jak i użytkownika, mają wiele znaczących różnic w stosunku do tych ostatnich. Przykładami kart T&E są *American Express Card* oraz *Dinners Club Card*. Polskimi przykładami są karty emitowane przez Bank Handlowy – *American Express*, PeKaO SA *VISA BUSINESS CARD*, czy BPH *VISA BUSINESS CARD*. Te ostatnie są wydawane polskim przedsiębiorcom posiadającym konta w oddziałach tego Banku.

Karta kredytowa (ang. *credit card*) pozwala na korzystanie ze środków, których jej posiadacz aktualnie nie ma na koncie, w formie kredytu gotówkowego udzielonego przez bank, lub też przez akceptanta w formie kredytu towarowego. Bank określa termin spłaty części lub całości płatności dokonanych kartą kredytową – najczęściej w ciągu 30 dni. Po przekroczeniu tego terminu kredytobiorca jest zobowiązany do spłacenia całości lub określonej części zaciągniętego kredytu. Częściowa wpłata jest traktowana jako spłata części limitu kredytowego oraz zaliczka na poczet odsetek.

Popularną metodą uatrakcyjnienia kart są bezpłatne dodatkowe świadczenia dla posiadaczy kart, takie jak ubezpieczenia w podróży, zniżki oraz pierwszeństwo w rezerwacjach hoteli. Karty kredytowe są ostatnio najczęściej łączone z innymi typami kart, tworząc w jednym kawałku plastiku kartę popularnie zwaną „wieloproduktową".

Pierwsza polska odmiana tego typu karty płatniczej, proponowana przez Glob Bank wspólnie ze spółką EKSPRESS M z Warszawy, pod nazwą *KARTA KREDYTOWA GLOB BANK – EKSPRESS M*, jest bardzo zbliżona do typowej karty kredytowej. Dzięki niewielkim opłatom wstępnym oraz przy konsumpcyjnym charakterze udzielanego kredytu (choć bardzo drogiego) ma ona zdecydowanie bardziej popularny charakter niż inne wydawane karty i jest prawdopodobnie najpopularniejszą dotychczas kredytową kartą płatniczą w Polsce.

Złote i platynowe wersje kart kredytowych i obciążeniowych – dostępne tylko wybranym klientom (najczęściej to bank wybiera klienta, a nie odwrotnie) powodują znaczne podniesienie prestiżu osoby posługującej się taką kartą, oferując większy niż w przypadku tradycyjnych wersji kart pakiet dodatkowych świadczeń oraz wyższe limity kredytowe.

Karty – narzędzia systemów bankowych

Karta gwarancyjna do czeków (ang. *cheque guarantee card*) identyfikuje Bank będący wydawcą karty i czeku oraz stanowi wzór podpisu posiadacza książeczki czekowej, któremu ją wydano. Karta gwarantuje płatność dokonaną czekiem do wysokości określonej na jej awersie. Czek potwierdzony kartą gwarancyjną (czego dokonuje przyjmujący czek przez wpisanie numeru karty na odwrocie czeku) staje się czekiem z gwarancją bezwarunkowej zapłaty. Przykładem jest karta gwarancyjna do czeków *EuroCheck*, proponowana już przez różne banki polskie, jak PeKaO SA czy BPH.

Karta bankomatowa, inaczej gotówkowa (ang. *Bankomat card, Bank Cash Card*) jest kartą służącą jako identyfikator posiadacza konta w sieci bankomatowej, pozwalającą na wypłatę gotówki oraz kontrolę stanu konta. Operacji kartą dokonuje się w automatach kasjerskich typu CD lub ATM, lub też bezpośrednio w kasie banku. Niekiedy występuje ona jako odrębne narzędzie, jednak najczęściej jest „wkomponowana" w inny produkt kartowy, taki jak: karta debetowa, kredytowa, a nawet gwarancyjna, tworząc kartę „wieloproduktową". Choć karta tego typu jest bardzo często traktowana jako karta debetowa, warto zauważyć, że w praktyce nie każda karta debetowa jest kartą bankomatową – i odwrotnie; nie każda karta gotówkowa jest kartą debetową – stąd też propozycja traktowania karty bankomatowej odrębnie i instrumentalnie.

Przykładem typowej karty bankomatowej jest wydana wspólnie przez szwedzkie banki – karta systemu *BANKOMAT*. Podobnie polskie karty: *PKO EKSPRES*, służące wypłatom w urządzeniach typu CD (*cash dispenser*) banku PKO BP na terenie całego kraju. Karty PeKaO SA – nazwane „Identyfikacyjną kartą magnetyczną" pozwalają również na wypłaty w kasach oddziału banku po podaniu kodu osobistego. Bank PBK wydał kartę *START* pozwalającą na pobieranie gotówki w oddziałach, w których zainstalowano bankomaty CD. Ostatnio do wydawców kart bankomatowych dołączyło wiele nowych banków, jak Bank Gdański, BPH oraz kilka banków spółdzielczych, np. BS w Kwidzynie czy BS w Legnicy.

Karta identyfikator pracownika banku – (*Id card*) służy w bankach do: kontroli dostępu do bankowych systemów komputerowych, weryfikacji dyspozycji pracowników banku, kontroli ruchu po pomieszczeniach banku, rejestracji czasu pracy itp. Karty te mają często naniesione zdjęcie osoby uprawnionej do jej używania, rzadko natomiast wzór jej podpisu. Jest to grupa instrumentów bankowych, w której najszerzej zastosowano już karty elektroniczne. W Polsce karty takie używane są już w większości banków.

Inne karty w systemach bankowych. Oprócz wyżej wymienionych – systemach bankowych mamy również do czynienia z wieloma innymi kartami o charakterze identyfikacyjnym i marketingowym. Na naszym rynku mamy przykład identyfikacyjnej karty promocyjnej Invest Banku o nazwie *Invest Club Member Card*, mającej charakter rabatowy, kart tabel denominacyjnych PeKaO SA czy wiele kart rabatowych i klubowych, wydawanych przez różne organizacje związane z bankami.

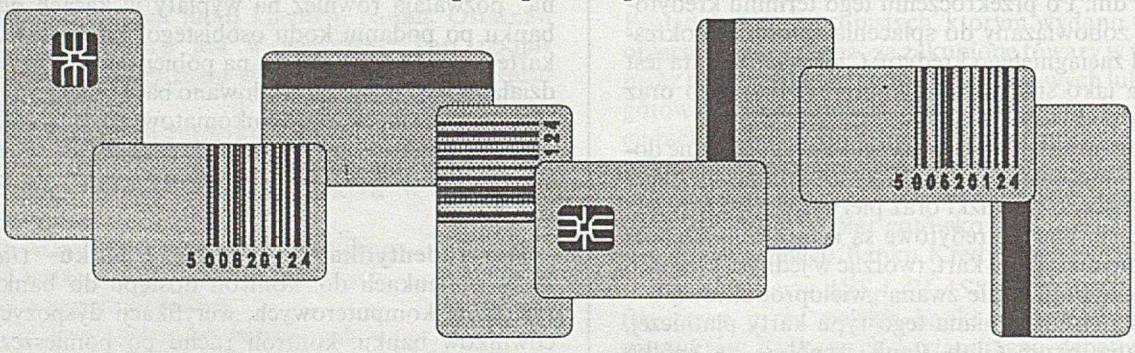
Karty te, choć zewnętrznie identyczne z kartami płatniczymi (wymiary, grafika, pasek magnetyczny), mają charakter bardziej marketingowy niż płatniczy, dlatego potraktowano je tu marginalnie.

* * *

Jak widać z tego krótkiego przeglądu – karty plastikowe w systemach bankowych to długa lista różnych zastosowań dla identycznego kawałka plastiku. Na naszym rynku dopiero zaczynają one się pojawiać, a już można zauważyć próby jednoczesnego wprowadzania wszystkich produktów kartowych. Od kart o charakterze ekskluzywnym – karty *AmEx* czy niektóre karty *VISA*, przez pierwsze karty bankomatowe banków PKO, BG, PBK oraz banków spółdzielczych, do popularnych kart Banku GLOB – *EKSPRESS M* czy BDK. Również obecne w Polsce oddziały banków zagranicznych proponują swoje produkty kartowe (Creditanstalt – karta *Diners Club*).

Na tym tle widać wyraźnie niespójność podejmowanych działań oraz brak wspólnej strategii naszych banków. Odnosi się wrażenie, że rynek kart płatniczych w Polsce jest zagospodarowywany nieplanowo i niespójnie. Brak rozbudowy niezbędnej do właściwego używania kart płatniczych infrastruktury (np. bazy kont osobistych), brak starań o odpowiednią edukację społeczeństwa oraz brak promocji produktów kartowych wywołują wrażenie, że poszczególne produkty kartowe pokazują się na rynku bardziej w celu podniesienia „prestiżu" wydawcy niż z potrzeby ekonomicznej. Przejawia się to, na przykład, powszechnym traktowaniem realizowanych przedsięwzięć kartowych pilotażowo i „eksperymentalnie". Polityka eksperymentów na ... przeciętnym obywatelu może niestety doprowadzić do stopniowego znużenia kolejnymi propozycjami, zanim uwierzy on w atrakcyjność produktów kartowych. Przykładem może być obserwowane ostatnio instalowanie pojedynczych bankomatów CD – notabene najdroższego z możliwych elementów systemów kart płatniczych, urządzeń służących w gruncie rzeczy do wypłacania gotówki. Właściwe uzasadnienie stosowania ATM i CD w krajach rozwiniętych, to możliwość korzystania z osobistego konta bankowego po godzinach pracy banku. Trudno taki argument zastosować w Polsce, gdzie niewiele ponad 10% społeczeństwa ma konta bankowe w ogóle, a wszyscy tradycyjnie jesteśmy przyzwyczajeni do regulowania wszelkich płatności gotówką. „Promocja" bankomatów przez brak możliwości posługiwania się kartą wydaną przez jeden bank w urządzeniach innych banków, a nawet sytuacja banków-wydawców kart, pozwalających na korzystanie z bankomatu tylko w oddziale, który wydał kartę i tylko w godzinach pracy banku, nasuwa poważne wątpliwości. Przy wysokim koszcie wstępnym instalacji bankomatu sprowadzenie go do funkcji normalnej kasy bankowej działającej w godzinach pracy banku jest zastanawiające. Analiza rynku wskazuje, że zainstalowane dotychczas CD są w ponad 95% instalacji urządzeniami do użytku wewnętrznego, a ich praca po zamknięciu banku jest ograniczana i okładana prowizjami. Jeżeli założyć we-

DIGITCARD - UNICARD
POLSKI INTEGRATOR SYSTEMÓW KARTOWYCH
proponuje:



projektowanie, produkcję i instalację systemów kart magnetycznych elektronicznych i kodów kreskowych
WYKORZYSTUJEMY POLSKĄ MYŚL TECHNICZNĄ I ŚWIATOWE KOMPONENTY
DIGITCARD - UNICARD sp. z o.o.
31-029 KRAKÓW UL. ZYBLIKIEWICZA 2/6, TEL: 22 00 16, 22 20 94, 23 11 62, FAX: 23 06 08
ODDZIAŁ: 00-656 WARSZAWA UL. ŚNIADECKICH 10/24 TEL/FAX: 621 72 15, 628 54 62
0/24/95

dług przyjętego w USA kanonu, że 75% wartości ATM powinno się zwrócić z opłat pobieranych za dokonanie wypłaty za pomocą tego urządzenia – trudno będzie przeciętnemu posiadaczowi konta w Polsce wytłumaczyć, po co mu karta gotówkowa. Dla przeciętnego obywatela poza „prestiżem" będzie to znaczyło tylko dodatkowe koszty, na które prawdopodobnie trudno będzie go namówić. Próby instalowania bankomatów w celu wypłat pensji też wydają się utrudnione, ponieważ dla średniej wielkości przedsiębiorstwa nie jest dziś możliwe przeprowadzenie wypłaty z jednego „załadunku" kaset CD starymi banknotami, a po całkowitym już wprowadzeniu nowych złotych sytuacja też nie poprawi się, ponieważ minimalna wypłata będzie wynosiła 10 lub 50 nowych zł!

Międzynarodowe karty płatnicze wydawane przez polskie banki w ramach międzynarodowych systemów kartowych, jak *VISA, Europay, Amex* stanowią dziś najlepiej rozwinięty i najpoważniejszy sektor rodzącego się rynku kartowego w Polsce. Wydawane przez BIG SA, PeKaO SA, BPH, WBK, BG, PBK, BŚ SA i inne, były początkowo skierowane do dobrze prosperujących firm i ich często podróżujących pracowników *VISA BUSINESS*, a ostatnio również do osób fizycznych *VISA CLASSIC*. Stanowią one bardzo cenny dokument podczas podróży do krajów rozwiniętych, gdzie załatwiają wszelkie problemy przewożenia, wymiany (choć nie po najkorzystniejszych kursach) i zapłat w obcych walutach wymienialnych. Propozycja ta wymaga niestety na ogół prowadzenia niskooprocentowanego konta walutowego (w USD) z określoną minimalną kwotą oraz równie nieatrakcyjnie oprocentowanego konta złotówkowego, co już czyni rynek tych kart raczej elitarnym i ograniczonym.

Karty tego typu stanowią bardzo ważny element systemu kartowego każdego cywilizowanego kraju – nie powinny być jednak nigdy jego głównym segmentem. Sytuacja powstała w naszym kraju może być traktowana przejściowo, przed wytworzeniem się narodowego systemu kartowego. Jest ona jednak o tyle cenna, że choć nie powoduje realnego upowszechnienia się kart płatniczych na naszym rynku, to wprowadza widoczny wzrost poziomu kultury bankowej oraz kultury używania kart płatniczych w ogóle. Bardzo cennym jest natomiast przenoszenie doświadczeń międzynarodowych systemów do organizacji systemów krajowych mające na celu wprowadzenie markowych produktów, jak np. podjęta ostatnio inicjatywa EUROPAY Int.

Innym obszarem do zagospodarowania jest krajowy rynek placówek handlowo-usługowych, gdzie karty płatnicze powinny znaleźć zastosowanie jako środek zapłaty w punktach sprzedaży i usług (zwanych akceptantami).

Potencjalnie istnieje bardzo duże zapotrzebowanie na usługi kartowe w tym sektorze rynku. Dowodzą tego pierwsze sukcesy w rozpowszechnianiu tego typu produktu kartowego przez Bank Depozytowo-Kredytowy z Lublina oraz Konsorcjum Finansowe EKSPRESS M z Warszawy. Rynek ten obsługiwany poprzednio przez Orbis, a potem POLCARD na rzecz międzynarodowych systemów kartowych, został poważnie „zmęczony" wysokimi prowizjami za obsługę transakcji oraz niską jakością usług (autoryzacji telefonicznej, czasem uznania, niskimi limitami). Według przeprowadzonych wstępnych sondaży akceptanci czekają na nowe produkty kartowe i nowe formy współpracy.

Przyszłość

Zarysowują się dwa kierunki rozwoju systemów kartowych w Polsce. Pierwszy z nich kładzie nacisk na stworzenie ogólnopolskiego systemu bankomatów, pracujących w sieci *on-line* i wprowadzenie kart gotówkowych. Przedsięwzięcie takie wymaga ogromnych nakładów oraz wieloletniego okresu wdrażania oraz „zestrajania" z istniejącymi systemami bankowymi, sposobami księgowania, działającymi bankomatami oraz warunkami prowadzenia wypłat. Efekt rynkowy natomiast mógłby być nieproporcjonalny do nakładów. Powszechna, w skali kraju, akceptacja społeczna urządzeń typu ATM czy CD mogłaby nastąpić dopiero po uruchomieniu całego ogólnopolskiego systemu autoryzacji, a jego nie dające się uniknąć przerwy w pracy (por. zawieszenie się całego systemu autoryzacyjnego CB na prawie 30 godzin we Francji w czerwcu 1994 r.) oraz prowizje od wypłat dodatkowo zniechęciłyby potencjalnych użytkowników kart gotówkowych.

Drugi kierunek jest związany z obsługą zakupów towarów i usług dzięki debetowym i kredytowym produktom kartowym dostępnym na terenie całego kraju. Niższe prowizje, lepsza obsługa, nowe formy usług dodanych, możliwość prowadzenia konta bezpośrednio u wydawcy kart, jak wykazuje wstępny marketing mogłyby spowodować większe zainteresowanie przyjmowaniem płatności kartami przez akceptantów. Z drugiej strony – realny wzrost liczby wydanych kart spowodowałby wzrost liczby prowadzonych kont osobistych oraz przeprowadzonych transakcji. Dobrym przykładem jest porównanie kart *EKSPRESS M* czy *BDK* do zaproponowanych na zbliżonych zasadach w tym samym czasie kart *POLCARD*. Potrzeby społeczeństwa znakomicie ilustruje liczba kart, wydanych przez wyżej wymienionych wydawców i będących w realnym obiegu. Prowadzi to do wniosku, że do czasu wytworzenia się niezbędnej infrastruktury lokalne karty płatnicze mają chyba obecnie większą szansę dynamicznego wejścia i rozwoju na naszym rynku płatniczym. Mają na to wpływ niskie koszty wprowadzenia systemu (kompletny system tradycyjnych kart debetowych dla pięćdziesięciotysięcznego miasta, obejmujący 500 sklepów jest tańszy, niż jeden ATM czy CD) oraz niskie koszty obsługi systemu z lokalną autoryzacją. System taki może być nową, ciekawą ofertą banku, polegającą na wprowadzeniu kart do istniejących już rachunków rozliczeniowo-oszczędnościowych czy kont osobistych. Dalsze wprowadzenie możliwości kredytowych (na istniejących już kartach) dla wybranych klientów banku może znaczyć udzielenie kilku tysięcy krótkoterminowych kredytów. O ile powstające systemy lokalne opierałyby się na podobnych zasadach formalnych oraz założeniach technicznych (na przykład wykonanie w technologii kart elektronicznych), to w niedalekiej przyszłości mogłyby one zostać połączone w ogólnopolski system kart płatniczych. Zdaniem autora, wytworzyłoby to docelowo bardzo spójny, atrakcyjny i potrzebny system kartowy dla całego kraju.

Alternatywą jest wprowadzenie tzw. markowych produktów kartowych, zaprojektowanych i wdrożonych wspólnie z *VISA INT* czy *EUROPAY*.

dokończenie na s. 31

Typy z protekcją w języku Ada '95

Zbigniew Fryźlewicz
Centrum Informatyczne
Politechnika Wrocławska

W artykule opisano nowy mechanizm synchronizacji zadań, wprowadzony w ramach modernizacji języka Ada i lepszego przystosowania go do tworzenia aplikacji silnie uwarunkowanych czasowo. Szczegółowo przedstawiono właściwości modelu, który leży u podstaw tego mechanizmu. Opisano instrukcję *requeue*, związaną z nowym mechanizmem synchronizacji oraz zastosowanie obu konstrukcji do budowy zadań pełniących rolę serwerów zasobów.

Interakcje między zadaniami w Adzie są realizowane w dwóch celach. Celem pierwszym jest komunikacja międzyzadaniowa. Celem drugim jest synchronizacja, w tym także synchronizacja dostępu zadań do zasobów dzielonych. W języku Ada '83 [6] te dwa zagadnienia są rozwiązywane na podstawie pojedynczego mechanizmu synchronicznych, asymetrycznych spotkań, nazywango *rendezvous*. Budowanie oprogramowania, w którym dostęp do danych dzielonych opiera się na mechanizmie *rendezvous*, nie jest rozwiązaniem najefektywniejszym czasowo. Przy tym mechanizmie potrzebne jest zawsze dodatkowe zadanie w postaci tzw. *managera zasobu*. W rezultacie w systemie mamy – oprócz zadań, które nadzorują współbieżne aktywności systemu – pewną liczbę zadań dodatkowych, z których każde pełni rolę managera jakiegoś zasobu. Powoduje to dodatkowe narzuty czasowe, związane z przełączaniem dodatkowych zadań.

W języku Ada '95 [1, 2, 5] utrzymany jest mechanizm *rendezvous* dla przesyłania komunikatów między zadaniami. Dodatkowo, dla synchronizacji dostępu do zasobów dzielonych [3], zdefiniowano nowy typ, tzw. *typ z protekcją*. Kontrolowany dostęp do zasobów dzielonych na podstawie obiekt tego typu jest – pod względem narzutów czasowych – podobnie efektywny, jak operacje semaforowe. Co więcej jest to mechanizm wysokopoziomowy i strukturalny.

Obiekt z protekcją

Forma zapisu obiektu z protekcją jest podobna do zapisu zadania i pakietu. Składa się z dwóch części, tj. specyfikacji i treści będącej implementacją obiektu. W ogólnym wypadku specyfikację tworzą deklaracje wejść, funkcji i procedur „eksportowanych" przez obiekt oraz część prywatna, wydzielająca zasoby dzielone. Dostęp do obiektu jest możliwy tylko przez wywołania eksportowanych wejść, funkcji i procedur. Dostęp, który modyfikuje obiekt lub składowe obiektu z protekcją i jest realizowany przez wywołania wejść lub procedur, działa wykorzystując mechanizm wzajemnego wykluczania. Dla dostępu typu „tylko odczytywanie", realizowanego przez wywołania eksportowanych funkcji, mechanizm wzajemnego wykluczania nie obowiązuje i odczytywać wartości obiektu może współbieżnie wiele zadań.

Analogicznie do monitora, obiekt z protekcją jest tworem statycznym. Konstrukcja ta ma także inne właściwości rozszerzające koncepcję monitora.

Najprostszy przykład obiektu z protekcją jest następujący:

```
protected ZMIENNA is
   function Czytaj return T;
   procedure Pisz (X: T);
private
   DANA: T;
end ZMIENNA;

protected body ZMIENNA is
   function Czytaj return T is
   begin
      return DANA;
   end Czytaj;

   procedure Pisz (X: T) is
   begin
      DANA := X;
   end Pisz;
end ZMIENNA;
```

W tym wypadku obiekt z protekcją zawiera zmienną *DANA* pewnego typu *T*. Zmienna ta może być odczytana lub zmodyfikowana, odpowiednio za pomocą funkcji *Czytaj* lub *Pisz*, przez dowolne współbieżnie działające zadanie. Stosowana jest w tym celu notacja kropkowa i przykładowe wywołania mają postać:

```
XX := ZMIENNA.Czytaj ;
. . .
ZMIENNA.Pisz(YY) ;
```

Wywołania *Czytaj* i *Pisz* działają na zasadzie wzajemnego wykluczania. Dodatkowo, wszystkie podprogramy typu *function* (w przykładzie *Czytaj*) mają prawo do odczytywania danych podanych w części *private* specyfikacji, a więc nie mogą tych danych modyfikować. Pozwala to na użyteczną optymalizację implementacyjną, gdyż wiele zadań może jednocześnie wykonywać taką funkcję.

Bariery

Podobnie jak zadania, obiekty z protekcją mogą w części specyfikacyjnej zawierać definicje wejść. W części implementacyjnej z każdym z zadeklarowanych wejść jest związana klauzula zakazu wykonywania wejścia nazywa-

na barierą. Treść wywoływanego wejścia jest wykonywana tylko wówczas, gdy warunek skojarzony z barierą przyjmuje wartość *TRUE*. Można zauważyć silną analogię między instrukcją *accept* z dozorami (konstrukcja w jednostkach typu *task*) a obiektami z protekcją i barierami. Istotna i zasadnicza różnica między tymi konstrukcjami jest związana z czasem obliczania warunków skojarzonych z dozorami i barierami.

Rozpatrzmy klasyczny przykład bufora cyklicznego. Odpowiednia definicja typu z protekcją ma postać:

```
protected type BUFOR is
  entry DAJ(x: out T);
  entry WEŹ(x: in T);
private
  buf: array(0..N-1) of T;
  w: Integer := 0;
  p: Integer := 0;
  licz: Integer range 0..N := 0;
end BUFOR;
protected body BUFOR is
  entry DAJ(x: out T) when licz > 0 is
    begin
      x := buf(p);
      p := (p + 1) mod N;
      licz := licz - 1;
    end DAJ;


  entry WEŹ(x: in T) when licz < N is
    begin
      buf(z) := x;
      z := (z + 1) mod N;
      licz := licz + 1;
    end WEŹ;
end BUFOR;
```

Deklaracja obiektu *M_BUF* typu *BUFOR* i wstawianie lub pobieranie elementów mają postać:

```
M_BUFOR: BUFOR;
Elem1, Elem2: T
. . .
M_BUFOR.WEŹ(Elem1);
. . .
M_BUFOR.DAJ(Elem2);
```

Treść, związana z wywoływanym wejściem jest przetwarzana tylko wówczas, gdy warunek skojarzony z barierą tego wejścia ma wartość *TRUE*. Warunek skojarzony z barierą jest obliczany, gdy wejście jest wywoływane po raz pierwszy i warunki barier – z niepustymi kolejkami zablokowanych zadań – są sprawdzane ponownie w chwili zakończenia wykonywania treści dowolnego wejścia lub procedury obiektu z protekcją. Warunki skojarzone z barierami nie są jednak obliczane z chwilą zakończenia funkcji. Funkcje mogą tylko odczytywać stan obiektu z protekcją, a więc nie zmieniają jego stanu, a zatem i warunków skojarzonych z barierami.

Z chwilą deklaracji bufora *buf* typu *M_BUFOR* jest on pusty i *licz* ma wartość 0. Przy braku wywołań wejścia *WEŹ*, zadania wywołujące wejście *DAJ* zostają zablokowane i ustawione w kolejce „na barierze" wejścia *DAJ*. Pierwsze zadanie, wywołujące wejście *WEŹ*, przechodzi przed barierę i wykonuje instrukcje zawarte w części implementacyjnej tego wejścia. W momencie zakończenia wykonywania tej części implementacyjnej warunki wszystkich barier, z niepustymi kolejkami zablokowanych zadań, są ponownie sprawdzane. Warunek skojarzony z barierą wejścia *DAJ* ma teraz wartość *TRUE* i pierwsze z zablokowanych na barierze zadań zostaje bezzwłocznie wznowione i wykona instrukcje zapisane w części implementacyjnej wejścia *DAJ*.

```
-- bufor N-elementowy
-- miejsce do wstawienia
-- miejsce do pobrania
-- wypełnienie bufora
```

Model szeregowania zadań

Model, opisujący zachowanie się zadań, próbujących „przedostać się przez barierę" jest pokazany na rys. 1. Można go nazwać **modelem skorupkowym** (ang. *the eggshell model*).

Z punktu widzenia obiektu z protekcją dowolne zadanie może być w jednym z trzech stanów:

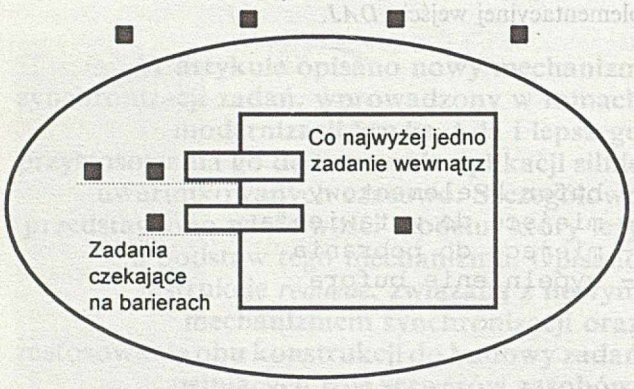
- wewnątrz obiektu, tzn. wykonywać instrukcje zawarte w części implementacyjnej pewnego wejścia,
- czekać w uporządkowanej kolejce związanej z jakąś barierą,
- być zadaniem „zewnętrznym", czekającym na możliwość obliczenia warunku skojarzonego z barierą.

Wewnątrz obiektu z protekcją może być co najwyżej jedno zadanie, które przeszło przez barierę. Gdy zadanie wewnątrz obiektu wykonuje instrukcje, zawarte w części implementacyjnej danego wejścia lub instrukcje treści podprogramu, wtedy mówimy, że obiekt jest „zajęty".

Jeżeli obiekt z protekcją nie jest „zajęty" to każde zadanie, wywołujące dowolne z wejść obiektu, powoduje obliczenie warunku skojarzonego z barierą.

Jeżeli wynikiem obliczenia warunku jest *FALSE*, to zadanie przechodzi przez pierwszy poziom protekcji i zostaje ustawione w kolejce zadań oczekujących na danej barierze. Istotne jest, że zadanie czeka teraz na przejście drugiego poziomu protekcji w uporządkowanej kolejce.

Jeżeli wynikiem obliczenia warunku jest *TRUE*, to zadanie przechodzi dwa poziomy protekcji i „zajmuje" obiekt. W tej samej chwili odcina całkowicie dostęp do tego obiektu zadaniom „z zewnątrz", uniemożliwiając im sprawdzanie warunków skojarzonych z barierami.



Dwupoziomowy model szeregowania zadań

Należy podkreślić, że mechanizm barier jest niezależny od warunku wzajemnego wykluczania, występującego z definicji w konstrukcji obiektu z protekcją. Tym samym mamy dwa różne poziomy protekcji. W momencie, gdy zadanie zajmujące obiekt z protekcją zwalnia go, wówczas warunki skojarzone z barierami są ponownie sprawdzane dla wszystkich zadań ustawionych w kolejkach na barierach. Zadania, dla których warunek ma teraz wartość *TRUE*, próbują zająć obiekt i mają w tym priorytet nad wszystkimi innymi wywołaniami, współzawodniczącymi o obiekt z protekcją. Z drugiej strony, jeżeli obiekt jest „zajęty" przez jakieś zadanie, to dopóki to zadanie nie zwolni obiektu (i nie zwolnią go ewentualne zadania, ustawione w kolejkach na barierach) to nowe wywołania wejść nie powodują nawet obliczania warunków skojarzonych z barierami.

Ma to ważne konsekwencje praktyczne. Jeżeli zmienia się stan obiektu z protekcją, i jeżeli jest zadanie czekające na nowy stan, to zadanie to uzyska dostęp do zasobu wraz z gwarancją niezmienniczości stanu obiektu między chwilą podjęcia decyzji o zwolnieniu obiektu (przez poprzednie zadanie) a chwilą uzyskania dostępu do obiektu. Nie ma możliwości hazardu, tj. sytuacji, że jakieś zadanie z zewnątrz „przeskoczy" zadania czekające w momencie zmiany stanu wewnątrz obiektu z protekcją.

Instrukcja requeue

Z obiektami z protekcją jest związana nowa instrukcja języka Ada 95 *requeue*. Może ona wystąpić tylko w części implementacyjnej wejścia obiektu z protekcją (lub w treści instrukcji *accept*). Zadanie, wykonujące *requeue*, umieszcza siebie ponownie w kolejce bariery, z której zostało odblokowane lub innej, kompatybilnej, kolejce.

Jak wykazuje praktyka programowania, rozwiązanie kilku modelowych problemów synchronizacyjnych, z zastosowaniem konstrukcji języka Ada 83, nastręcza trudności. W szczególności dotyczy to zadań pełniących rolę serwerów zasobów [7].

Rozpatrzmy zaprogramowanie następującego, dwustopniowego modelu synchronizacji zadań. Grupa zadań synchronizuje się w oczekiwaniu na pewne cyklicznie zachodzące zdarzenie o charakterze sygnału *broadcast*. Każde z zadań zgłasza gotowość do synchronizacji wywołaniem:

Zdarzenie.Czekaj

Wystąpienie sygnału *broadcast* jest sygnalizowane przez jedno z zadań wywołaniem:

Zdarzenie.Można_wznowić

Wtedy wszystkie zadania, które czekały na *broadcast* wznawiają swoją asynchroniczną aktywność i system jest powtórnie gotowy do synchronizacji.

Trudość polega na tym, by przez tak zdefiniowany punkt synchronizacyjny nie „prześlizgnęły" się zadania, które zgłosiły gotowość do czekania na zdarzenie po tym, kiedy ono już wystąpiło, ale jeszcze przed wyzerowaniem znacznika (flagi) o jego wystąpieniu.

Rozwiązanie tego problemu, z zastosowaniem obiektu z protekcją i instrukcją *requeue*, jest przedstawione na wydruku na sąsiedniej stronie.

Zmienna bulowska *Wystąpiło* ma stale wartość *False* i przybiera wartość *True* tylko w chwili uwalniania zadań zablokowanych w kolejce na barierze. Wejście *Czekaj* ma w części implementacyjnej tylko instrukcję pustą. Ta konstrukcja powoduje, że wszytkie zadania, wywołujące wejście *Czekaj* są blokowane w kolejce na barierze do momentu, gdy zmienna *Wystąpiło* przybierze wartość *True*. Warunek skojarzony z barierą wejścia *Można_wznowić* ma stale wartość *True*. Jeżeli jest ono wywołane i brak jest zadań zablokowanych na barierze wejścia *Czekaj*, to nie są wykonywane żadne instrukcje i przetwarzanie tego wejścia kończy się (*broadcast* wystąpił, zanim jakiekolwiek zadanie zadeklarowało gotowość do czekania; efekt jest pusty). Z drugiej strony, jeżeli w kolejce na barierze wejścia *Czekaj* są zablokowane jakieś zadania, to muszą one zostać zwolnione, i to w taki sposób, aby żadne nowe zadanie „zewnętrzne" nie pojawiło się w kolejce bariery tego wejścia *po* momencie wywołania wejścia *Można_wznowić*. Krytyczne dla poprawnego rozwiązania jest odzyskanie sterowania dla wyzerowania znacznika *Wystąpiło*. Opisana sekwencja akcji jest osiągana za pomocą instrukcji *requeue*.

Z definicji działania obiektu z protekcją wynika, że z chwilą opuszczania tego obiektu są na nowo sprawdzane warunki, skojarzone z barierami tych wejść, w których są zablokowane jakieś zadania. W rozważanym wypadku mamy zadania na barierze wejścia *Czekaj* i na barierze wejścia *Zeruj* (zadanie, które wywołało wejście *Można_wznowić*). Warunek, skojarzony z barierą wejścia *Czekaj*, ma wartość *True*. Ponieważ są zadania w kolejce wejścia *Czekaj*, to warunek skojarzony z barierą wejścia *Zeruj* ma wartość *False*. Jedno z czekających zadań jest zwalniane i wykonuje instrukcje z części implementacyjnej wejścia *Czekaj*. W istocie jest to tylko instrukcja pusta, zatem zadanie zwalnia obiekt z protekcją, ale powoduje też ponowne obliczenie warunków barier wszystkich wejść z zablokowanymi zadaniami. Powyższa sekwencja akcji powtarza się, do momentu aż uwolnione zostaną wszystkie zadania czekające w kolejce bariery wejścia *Czekaj*. Dopiero wtedy warunek skojarzony z barierą wejścia *Zeruj* przyjmuje wartość *True*. W tym momencie zadanie, które początkowo wywołało wejście *Można_wznowić* i na skutek działania *requeue* zostało prze-

```
protected Zdarzenie is
  entry Czekaj;
  entry Można_wznowić;
private
  entry Zeruj;
  Wystąpiło: Boolean := False;
end Zdarzenie;

protected body Zdarzenie is
  entry Czekaj when Wystąpiło is
  begin
    null;
  end Czekaj;
  entry Można_wznowić when True is
  begin
    if Czekaj'Count >0 then   -- sprawdź czy kolejka niepusta
      Wystąpiło := True;
      requeue Zeruj;
    end if;
  end Można_wznowić;
  entry Zeruj when Czekaj'Count = 0 is
  begin
    Wystąpiło := False;
  end Zeruj;
end Zdarzenie;
```

mieszczone i zablokowane na barierze wejścia *Zeruj*, jest odblokowywane i wykonuje instrukcje części implementacyjnej tego wejścia. Ustawia ono znacznik *Wystąpiło* na *False* i tym samym przywraca stan początkowy systemu i obiektu z protekcją *Zdarzenie*.

Zauważmy, że jeżeli jakieś zadania wywoływały wejścia *Czekaj* i (lub) *Można_wznowić* w czasie, gdy obiekt z protekcją *Zdarzenie* był zajęty opisanym wyżej przetwarzaniem, to żadne z tych wywołań nie spowodowało nawet obliczenia warunku skojarzonego z odpowiednią barierą. Wszystkie te zadania oczekują w niestrukturalnej „kolejce" na sposobność obliczeń warunków skojarzonych z barierami. Może to nastąpić dopiero po całkowitym zwolnieniu obiektu przez poprzednio zajmujące go zadania. Podany przykład jest ilustracją podwójnej „ochrony" obiektu z protekcją i pokazuje mechanizmy zapobiegania hazardom.

Wprowadzenie obiektów z protekcją do języka Ada jest istotną jego modernizacją. Obiekty z protekcją łączą w sobie zalety mechanizmu synchronizacji, zwanego monitorem, z zaletami strukturalnego mechanizmu synchronizacji na podstawie modelu spotkań z dozorami. Są konstrukcjami statycznymi i stąd efektywność czasowa przy ich implementacji – co ma istotne znaczenie przy tworzeniu aplikacji silnie uwarunkowanych czasowo. Podwójna protekcja eliminuje zjawisko hazardu, które występowało w szczególnych okolicznościach przy synchronizacji zadań w klasycznej Adzie 83. Wreszcie jest to mechanizm strukturalny. Poprawność konstrukcji jest łatwiejsza do udowodnienia, gdyż jest opisana przez bariery, a nie przez niskopoziomowe i niestrukturalne instrukcje typu *signal* i *wait*, występujące w klasycznych monitorach.

LITERATURA

[1] Ada 9X Requirements. Office of the Under Secretary of Defence for Acquisition, Washington, D.C., 1990

[2] Ada 9X Requirements Rationale. Office of the Under Secretary of Defence for Acqusition, Washington, D.C., 1991

[3] Dewar R.B.K.: Shared Variables and Ada 9X Issues. Special Report SEI-90-Sr-1. Software Engineering Institute. January 1990

[4] Hilzer R.C.: Synchronisation of the Producer/Consumer Problem using Semaphores, Monitors, and the Ada Rendezvous. ACM Operating Systems Review, 26 (3), July 1992

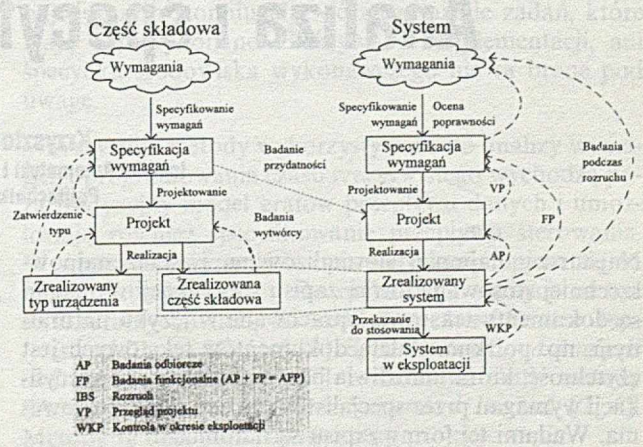
[5] International Standards Organisation. Reference Manual for the Ada Programming Language. ISO/8652- 1995, 1995

[6] Reference Manual for the Ada Programming Language. ANSI/MIL-Std-1815a edition, 1983

[7] Wellings A.J., Keeffe D., Tomlison G.M.: A Problem with Ada and Resource Allocation. ACM SIGAda Ada Letter: 112–113, 1984

UZUPEŁNIENIE

W sierpniowym numerze INFORMATYKI we wkładce „Systemy czasu rzeczywistego" w artykule „Atestacja, certyfikacja i licencjonowanie systemów komputerowych pełniących funkcje związane z bezpieczeństwem" zamieściliśmy niekompletny rysunek oznaczony numerem 3. Obok powtarzamy go w poprawnej formie wraz z podpisem.



Rys. 3

Związek pomiędzy zatwierdzeniem typu i zatwierdzeniem zależnym od zastosowania →

SYSTEMY CZASU RZECZYWISTEGO

REDAGUJE ZESPÓŁ: dr inż. Zbigniew FRYŹLEWICZ (Politechnika Wrocławska), prof. dr hab. inż. Janusz GÓRSKI (Francusko-Polska Wyższa Szkoła Nowych Technik Informatyczno-Komunikacyjnych w Poznaniu), prof. dr hab. inż. Zbigniew HUZAR (Politechnika Wrocławska), dr inż. Krzysztof SACHA (Politechnika Warszawska), prof. dr hab. inż. Tomasz SZMUC (Akademia Górniczo-Hutnicza), dr inż. Zbigniew ZIELIŃSKI (Wojskowa Akademia Techniczna), mgr inż. Zdzisław ŻURAKOWSKI (Instytut Automatyki Systemów Energetycznych we Wrocławiu) – przewodniczący zespołu.

ROK II ● PAŹDZIERNIK ● 1995 ● 11

Każdy proces rozwoju oprogramowania obejmuje cztery rodzaje działań: określenie potrzeb, znalezienie rozwiązania, realizację i ocenę rezultatów. Działania te następują na ogół po sobie, tworząc kolejne etapy cyklu projektowego:

- analizę i specyfikację wymagań,
- projekt wstępny i szczegółowy,
- implementację (programowanie),
- integrację, weryfikację i ocenę.

Krótką charakterystykę poszczególnych etapów cyklu projektowego oraz opis najważniejszych problemów i metod ich rozwiązania można znaleźć w przeglądowym artykule [4], wymienionym w wykazie literatury do artykułu publikowanego w tym numerze.

Publikowany niżej artykuł otwiera cykl poświęcony przedstawieniu metod analizy i specyfikacji wymagań dla oprogramowania systemów czasu rzeczywistego.

Głównymi celami etapu analizy i specyfikacji wymagań są: zrozumienie problemu, przed którym stanął użytkownik (analiza) i zakomunikowanie tej wiedzy zespołowi projektującemu programy (specyfikacja). Zasadniczy schemat postępowania polega na dekompozycji zagadnienia na fragmenty, mniejsze i łatwiejsze do zrozumienia i opisania. Stosowane w tym celu metody powinny dostarczyć modelu dekompozycji oraz określić sposób jednoznacznego zapisu wymagań. Na obecnym etapie rozwoju technologii oprogramowania dominującą rolę odgrywają dwie grupy metod: analizy strukturalnej, oparte na modelu przepływu danych i analizy obiektowej, oparte na modelu struktury i zależności obiektów występujących w dziedzinie zastosowania.

Każda z istniejących metod ma pewne zalety i wady. Czynnikiem, utrudniającym wiarygodną i jednoznaczną ocenę, jest istotna sprzeczność wymagań stawianych metodom analizy i specyfikacji. Z jednej strony, metody te powinny być zrozumiałe i poglądowe, tak aby ułatwić analizę problemu i komunikację analityków z użytkownikami. Z drugiej strony, powinny być ścisłe i formalne, aby umożliwić jednoznaczny zapis wymagań, wykluczyć możliwość różnych interpretacji i umożliwić dowodzenie poprawności. Pewnym rozwiązaniem tej sprzeczności jest szerokie wykorzystanie narzędzi graficznych, łączących poglądowość z precyzyjnie opracowaną składnią i semantyką.

Zadaniem rozpoczynanego w tym numerze cyklu artykułów jest przedstawienie najważniejszych nurtów, występujących w praktyce analizy i specyfikacji wymagań. W ramach tego cyklu zamierzamy przedstawić kolejno:

- przegląd dziedziny (artykuł w tym numerze);
- metody analizy strukturalnej;
- metody analizy obiektowej;
- graficzne specyfikacje w systemie La4.

K.S.

Metody analizy i specyfikacji wymagań

Analiza i specyfikacja wymagań

Krzysztof M. Sacha
Instytut Automatyki i Informatyki Stosowanej
Politechnika Warszawska

Najstarszą, najmniej sformalizowaną i wciąż najpowszechniej stosowaną formą zapisu specyfikacji wymagań są dokumenty tekstowe, opracowane w języku naturalnym, np. polskim. Zaletą dokumentów tekstowych jest czytelność, która umożliwia bezpośrednią ocenę specyfikacji wymagań przez specjalistów z dziedziny zastosowania. Wadami tej formy zapisu są natomiast:

- niejednoznaczność, wynikająca nieuchronnie z niejednoznaczności języka naturalnego;
- brak możliwości analizy i dowodzenia niesprzeczności specyfikacji;
- niska wiarygodność oceny specyfikacji, która w ogromnym stopniu zależy od wyobraźni użytkownika.

Podstawowym sposobem zwiększenia jednoznaczności specyfikacji jest formalizacja zapisu, prowadząca do wprowadzenia sztucznych języków o dokładnie zdefiniowanej składni i semantyce, nie dopuszczających możliwości różnej interpretacji. Prace nad formalizacją zapisu specyfikacji toczą się w dwóch kierunkach [8]: metod matematycznych, zorientowanych na dowodzenie i języków wykonalnych, zorientowanych na badanie eksperymentalne.

Metody matematyczne odwołują się na ogół do teorii abstrakcyjnych typów danych i próbują zapisać specyfikację w formie zespołu warunków logicznych, zbudowanych nad zbiorami reprezentującymi wartości danych. Zaletami specyfikacji matematycznych są:

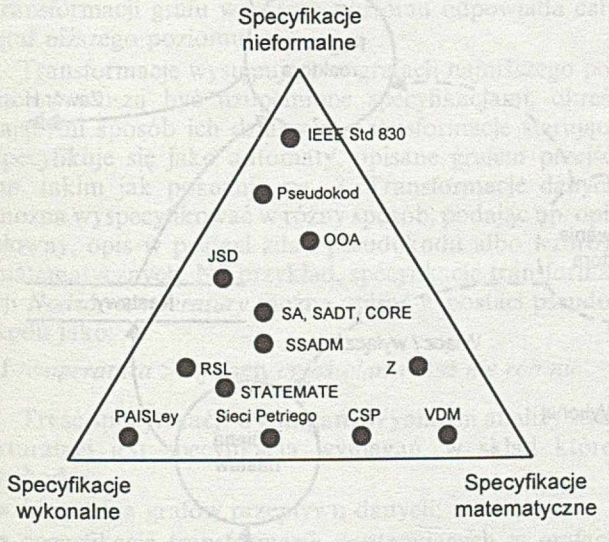
- jednoznaczność zapisu;
- możliwość dowodzenia niesprzeczności specyfikacji i poprawności programu.

Do wad należą mała czytelność i wynikająca z niej bardzo niska wiarygodność oceny poprawności specyfikacji wymagań, dokonywanej przez użytkownika.

Specyfikacje wykonalne opierają się z reguły na wykorzystaniu języków bardzo wysokiego poziomu, np. języków funkcjonalnych, listowych lub grafowych, które umożliwiają szybką budowę programu o znikomej efektywności wykorzystania zasobów. Opracowana w formie programu specyfikacja jest wykonalna i stanowi makietę docelowego oprogramowania, która może być użyta do eksperymentalnego badania jego spodziewanych właściwości. Warto zauważyć, że wykonalna specyfikacja jest również w pełni formalna i jednoznaczna, co umożliwia matematyczne badanie i dowodzenie jej właściwości. Zaletami specyfikacji wykonalnych są:

- jednoznaczność zapisu;
- możliwość dowodzenia niesprzeczności specyfikacji;
- możliwość eksperymentalnego badania makiety programu.

Wadą jest mała czytelność zapisu, utrudniająca ocenę tych właściwości, które nie mogą być zbadane analitycznie lub pokazane podczas wykonania makiety.



Rys. 1. Klasyfikacja metod specyfikacji wymagań

W praktyce specyfikowania wymagań na oprogramowanie systemów czasu rzeczywistego dominują metody pośrednie, wykazujące pewien stopień formalizacji, lecz dalekie od rygorów matematycznej ścisłości. Metody te nie są również wykonalne, ale w większości wypadków przewidują możliwość szybkiej i taniej budowy makiety programu, która może być zbadana eksperymentalnie, lub przedstawiona do oceny użytkownikowi. Orientacyjną klasyfikację wielu znanych metod specyfikacji wymagań, zestawioną ze względu na stopień formalności i wykonalności, przedstawia rys. 1.

Metody zapisu – formalne lub nie – służą jedynie specyfikowaniu i dokumentowaniu wymagań. Osobnym problemem, który musi być rozwiązany wcześniej, jest analiza wymagań, tzn. ustalenie tego, co projektowane oprogramowanie ma wykonywać. Podstawowym schematem analizy jest dekompozycja całości problemu na mniejsze, łatwiejsze do zrozumienia fragmenty. Większość metod specyfikacji wymienionych na rys. 1 i stosowanych w praktyce opiera się na wykorzystaniu jednego z trzech modeli dekompozycji:

- przepływu danych (metody analizy strukturalnej – SA, SADT, CORE, SSADM);
- struktury obiektowej (metody analizy obiektowej – różne warianty OOA);
- procesów współbieżnych (PAISLey, CSP, JSD, niektóre rodzaje sieci Petriego).

W dalszej części pracy zostanie przedstawiony krótki przegląd zasadniczych cech metod analizy i specyfikacji wymagań, opartych na tych modelach.

Analiza strukturalna

Grafowe modele przepływu danych stanowią jeden z najstarszych sposobów przedstawiania opisu przetwarzania informacji, stosowany tradycyjnie do opisywania systemów administracyjnych i bankowych. Wykorzystanie tych metod do modelowania oprogramowania doprowadziło do powstania dużej grupy metod, obejmowanych zbiorczą nazwą analizy i projektowania strukturalnego. Wszystkie metody tej grupy opierają się na zstępującym podejściu do projektowania oprogramowania i są wkomponowane w proces projektowy oparty na modelu kaskadowym.

Metody analizy strukturalnej budują mniej lub bardziej sformalizowany *model funkcjonalny* projektowanego systemu. Model składa się z grafu przepływu danych, pokazującego wymagane funkcje przetwarzania danych, uzupełnionego opisem szczegółów działania funkcji. Struktura modelu jest dobierana w taki sposób, aby w jak największym stopniu ułatwić zrozumienie zadań, które mają być wykonane. Efektywność implementacji, ani specyfika środowiska wykonawczego nie są brane pod uwagę.

Wszystkie metody wykorzystywane do analizy wymagań oprogramowania czasu rzeczywistego wychodzą poza tradycyjny model grafów przepływu danych i umożliwiają również specyfikowanie przepływu sterowania. W większości wypadków opis sterowania opiera się na modelu automatowym. Do najbardziej znanych metod strukturalnych należą: SSADM (*Structured System Analysis and Design Method*) [3], SADT (*Structured Analysis and Design Technique*) [7] i różne warianty metody SA/SD (*Structured Analysis/Structured Design*) [2, 6]. Metody te obejmują również etap projektowania.

Przykład

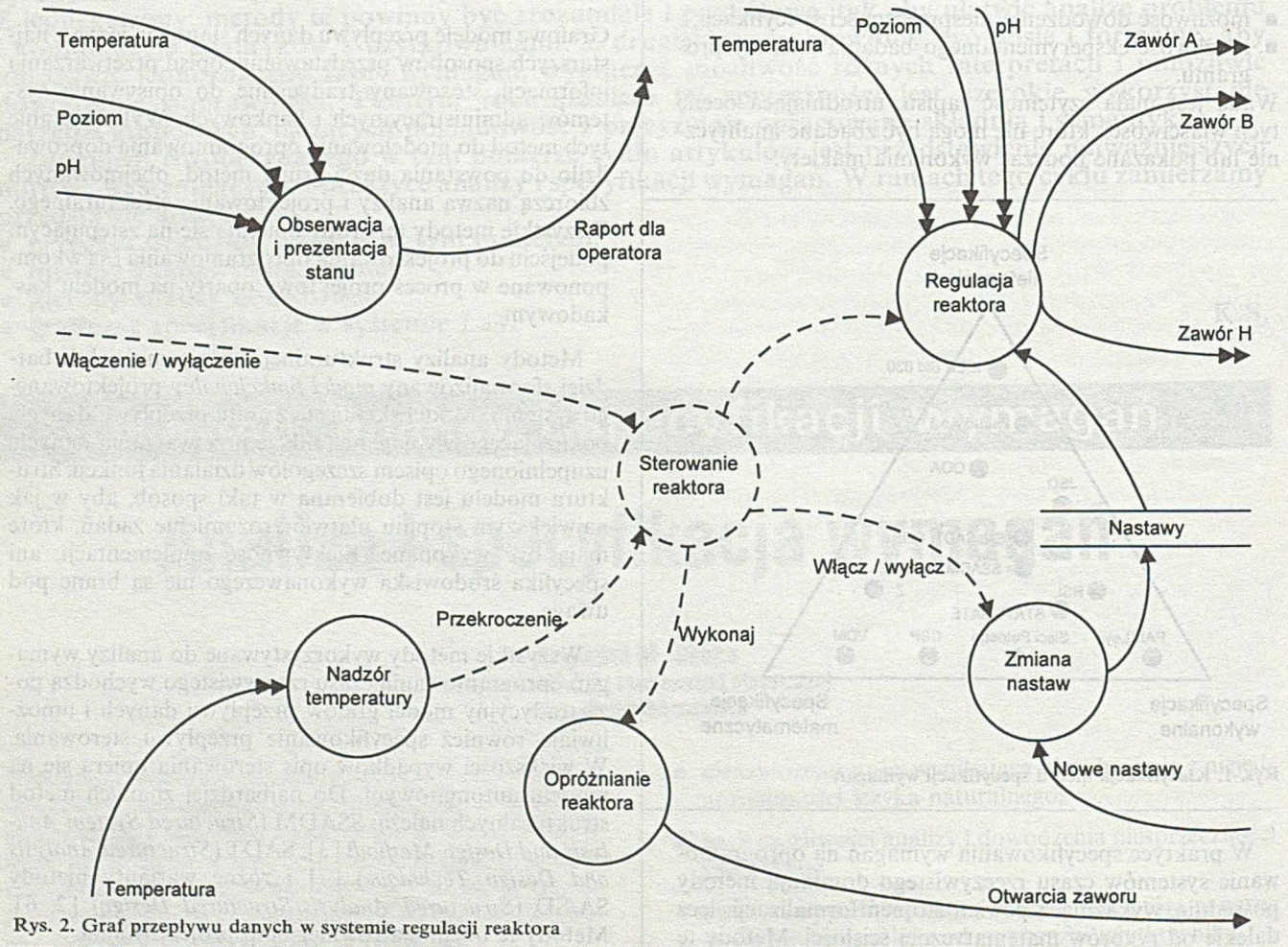
Rozważmy problem analizy i specyfikacji wymagań dla oprogramowania sterującego reaktorem chemicznym. Wymagania użytkownika obejmują stabilizację temperatury, pH i poziomu cieczy oraz ciągłą obserwację stanu procesu i prezentację operatorowi. Ponadto program powinien nadzorować temperaturę i powodować natychmiastowe opróżnienie reaktora w wypadku przekroczenie poziomu alarmowego.

Dekompozycja wymagań i związanego z nim przetwarzania polega na wyodrębnieniu poszczególnych czynności przetwarzania, nazywanych transformacjami danych i połączeniu ich strzałkami, pokazującymi przepływy danych niezbędne dla współpracy transformacji. W tym przykładzie podstawowymi czynnościami (transformacjami) są obserwacja i prezentacja stanu reaktora, regulacja stanu reaktora, ustawienie nastaw regulatorów, oraz nadzór temperatury i sterowanie awaryjnym opróżnieniem reaktora.

Zgodnie z notacją, zaproponowaną przez Warda i Mellora w [7], graf przepływu danych składa się z okręgów – reprezentujących transformacje, otwartych prostokątów – reprezentujących zbiory lub struktury danych oraz skierowanych łuków – reprezentujących przepływ danych lub przepływ sterowania (rys. 2). Łuki grafu zakończone pojedynczymi strzałkami oznaczają przepływy dyskretne, których wartości mogą być odczytane w określonych chwilach czasu. Łuki wyróżnione podwójnymi strzałkami oznaczają przepływy ciągłe, których wartości są zawsze dostępne i mogą być odczytane w dowolnym momencie. Zawsze są również dostępne dane zapisane w zbiorach.

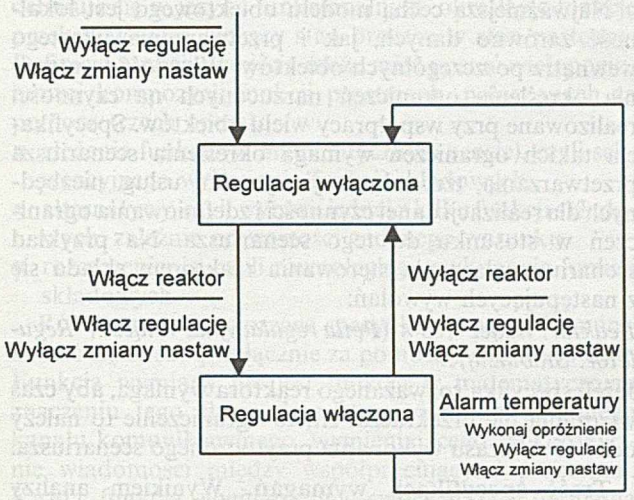
Przepływy dyskretne można uważać za wiadomości przenoszące wartości zmiennych i powodujące wykonanie transformacji, do których są skierowane. Przepływy dyskretne mogą również powodować zapamiętywanie danych w zbiorach. Na przykład, na rys. 2 transformacja *Zmiana nastaw* wykonuje się w chwili pojawienia się zewnętrznego przepływu, przenoszącego *Nowe nastawy* podane przez operatora. Wykonanie tej transformacji powoduje zapamiętanie nowej wartości w zbiorze *Nastawy*. Transformacje przetwarzające zmienne ciągłe (np. temperaturę) działają w sposób ciągły. Szczegóły realizacyjne, w tym konieczność kwantyzacji wartości i dyskretyzacji czasu są na poziomie specyfikacji wymagań nieistotne.

Zgodnie z logiką grafów przepływu danych transformacje, które mają określone wszystkie przepływy wejściowe, produkują przepływy wyjściowe niezależnie od działania innych transformacji. Nie zawsze jest to dopuszczalne z punktu widzenia opisywanego zastosowania. Na przykład, w rozważanym systemie sterującym reaktorem chemicznym wymagania użytkownika wykluczają możliwość zmiany nastaw regulatorów podczas pracy systemu – zmiana nastaw może spowodować zmianę jakości produktu reakcji, co nie może następować podczas biegu procesu. Tego rodzaju zależności między transformacjami regulują wydzielone transformacje sterujące, które mogą włączać lub wyłączać inne transformacje, tak aby



Rys. 2. Graf przepływu danych w systemie regulacji reaktora

zapewnić pożądaną kolejność działań. Na rysunku 2 taką rolę pełni dodatkowa transformacja sterująca *Sterowanie reaktora*. Transformacje i przepływy sterujące są rysowane liniami przerywanymi.



Rys. 3. Graf przejść funkcji *Nadzór systemu*

Działanie transformacji sterującej można opisać za pomocą grafu przejść automatu (rys. 3). Elementami grafu są prostokąty, odpowiadające stanom i łuki, obrazujące przejścia między stanami. Opis łuku składa się z dwóch elementów, oddzielonych poziomą kreską: zdarzenia (przepływu wejściowego transformacji), które powoduje przejście z jednego stanu do drugiego i akcji, która ma być wykonana.

W miarę postępu analizy transformacje, opisujące wymagane funkcje, podlegają dekompozycji na zespoły prostszych transformacji, połączonych przez przepływy i zbiory. Ponieważ liczba szczegółów, jakie można przedstawić na pojedynczym rysunku jest ograniczona, konieczna staje się reprezentacja hierarchiczna, w której jednej transformacji grafu wyższego poziomu odpowiada cały graf niższego poziomu.

Transformacje występujące w grafach najniższego poziomu muszą być uzupełnione specyfikacjami, określającymi sposób ich działania. Transformacje sterujące specyfikuje się jako automaty, opisane grafem przejść, np. takim jak pokazuje rys. 3. Transformacje danych można wyspecyfikować w różny sposób, podając np. opis słowny, opis w postaci zdań pseudokodu albo wzorów matematycznych. Na przykład, specyfikację transformacji *Nadzór temperatury* można opisać w postaci pseudokodu jako:

if *temperatura* $> t_0$ **then** *ogłoś alarm* **else** *nie rób nic*

Treść specyfikacji wymagań. Wynikiem analizy strukturalnej jest specyfikacja wymagań, w skład której wchodzą:

- hierarchia grafów przepływu danych;
- specyfikacje transformacji, występujących w grafach najniższego poziomu;
- opis danych, gromadzonych w zbiorach i przepływających między transformacjami;
- opis wymagań niefunkcjonalnych, dotyczących żądanej szybkości, dokładności i wydajności oprogramowania.

Analiza obiektowa

Naturalną reprezentację niemal każdej dziedziny zastosowania stanowi zbiór obiektów występujących w realnym świecie, uzupełniony regułami współdziałania tych obiektów. Na przykład, w systemie graficznego sprzęgu komputera z użytkownikiem występują obiekty, takie jak: okienka, okienka dokumentów, okienka menu, myszka itp. Reguły działania systemu wyznacza oddziaływanie myszki na okienka menu i funkcji menu na pozostałe okienka. Podobnie w instalacji chemicznej występują obiekty, takie jak czujniki, regulatory i zawory nastawiające, a reguły działania wyznaczają oddziaływania czujników na regulatory i regulatorów na zawory nastawiające.

Metody obiektowe starają się odwzorować strukturę obiektów i ich połączeń w rzeczywistym świecie na strukturę obiektów programowych. Każdy obiekt programowy jest opisany przez atrybuty, czyli dane definiujące stan obiektu oraz usługi, czyli operacje wykonywane na tych danych. Na przykład, atrybutami regulatora mogą być nastawy, wartość zadana, numer czujnika mierzącego wartość bieżącą i numer zaworu nastawiającego, a usługami – *zmień nastawy* i *stabilizuj*. Atrybuty i usługi stanowią nierozerwalną całość, w tym sensie, że usługi obiektu odnoszą się tylko do jego atrybutów, a wartości atrybutów obiektu mogą być zmieniane tylko przez jego usługi. Jedynym sposobem komunikowania się obiektów jest wywoływanie usług innych obiektów.

Obiekty, mające te same atrybuty i usługi tworzą klasę obiektów. Każda klasa określa pewien wzorzec struktury, zgodnie z którym można powoływać dalsze obiekty tego samego rodzaju. Celem analizy jest zidentyfikowanie klas obiektów, znajdujących się w polu widzenia opracowywanego programu i opisanie wszystkich zależności, istniejących między obiektami należącymi do tych klas. Zależności te mogą mieć charakter statycznych relacji, typu całość–część, np. pętla regulacyjna i regulator, lub generalizacja–specjalizacja, np. czujnik i czujnik temperatury albo oddziaływań dynamicznych, polegających na wywołaniu usług, np. *wykonaj pomiar*. Wynikiem analizy jest model dziedziny zastosowania zbudowany z klas obiektów, które reprezentują elementy rzeczywistości oraz połączeń obiektów, które pokazują zależności i interakcje zachodzące między obiektami.

Przykład

Rozważmy model obiektowy (rys. 4), opisujący strukturę sterowania reaktorem chemicznym rozważanym w rozdz. 2. Elementami modelu są klasy obiektów rzeczywistych: *Reaktor, Czujnik, Czujnik temperatury, Zawór, Zawór spustowy* i abstrakcyjnych: *Regulator* i *Pętla Regulacyjna*. Obiekty abstrakcyjne uosabiają pojęcia, którymi posługują się specjaliści z dziedziny zastosowania modelowanego systemu, tu – automatyki przemysłowej.

Relacja całość-część jest zaznaczona na rysunku linią ciągłą, z dopisaną obok liczbą części; w skład każdej pętli regulacyjnej wchodzi więc jeden czujnik, jeden regulator i jeden zawór. Warto zauważyć, że pętle regulacyjne: temperatury, poziomu i pH nie są rzeczywistymi częściami reaktora, ale stanowią abstrakcje działań, które składają się na całość funkcjonowania reaktora. Relacja generalizacja-specjalizacja jest oznaczoną linią z dodatkowym

symbolem półokręgu. Jest to notacja podobna do zaproponowanej przez Coada i Yourdona w [1]. Obiekty wywołują wzajemnie swoje usługi, w sposób pokazany na rysunku za pomocą strzałek. Zestaw wszystkich usług wykonywanych przez obiekty modelu jest zebrany w tab. 1.

Obiekty i operacje

Obiekt	Usługi	Usługi wywoływane
Reaktor	Włącz Wyłącz	Pętla Regulacyjna. Włącz Pętla Regulacyjna. Wyłącz
Pętla Regulacyjna	Włącz Wyłącz	Regulator. Stabilizuj Regulator. Wyłącz
Regulator	Ustaw nastawy Stabilizuj Wyłącz	 Czujnik. Podaj wartość Zawór. Ustaw przepływ
Czujnik	Podaj wartość	
Czujnik temperatury	Podaj wartość Nadzoruj	 Reaktor. Wyłącz Zawór spustowy. Ustaw otwarty
Zawór	Ustaw zawór	
Zawór spustowy	Ustaw otwarty Ustaw zamknięty	

Zbudowanie modelu obiektowego pozwala uchwycić strukturę dziedziny zastosowania i pokazać zależności występujące między istniejącymi w tej dziedzinie obiektami. Opis każdego obiektu musi być uzupełniony szczegółowymi specyfikacjami atrybutów oraz specyfikacjami usług. Metody specyfikowania atrybutów i usług nie różnią się istotnie od analogicznych metod specyfikowania danych i transformacji w modelu grafu przepływu danych analizy strukturalnej. Przykładowa specyfikacja usługi *Nadzoruj* obiektu *Czujnik temperatury* może mieć postać:

Permanentnie:

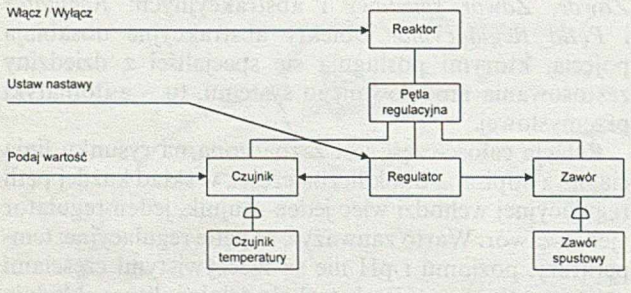
if *temperatura* $> t_0$

then (Reaktor. Wyłącz **and** Zawór spustowy. Ustaw otwarty)

else *nie rób nic*

Warto zauważyć, że usługa *Nadzoruj* określa permanentny, cykliczny proces kontroli temperatury, jest więc czymś więcej niż tylko „operacją" nad atrybutami obiektu. Pokazuje to, że modele obiektowe są istotnie ogólniejsze od modeli abstrakcyjnych typów danych, wykorzystywanych przez metody specyfikacji matematycznych.

Atrybuty i usługi nie są wystarczającymi narzędziami do pełnego opisania wszystkich aspektów działania realnych obiektów. Brakującym elementem jest opis wewnętrznego sterowania, definiujący pewien model „stanów życia" obiektu. Model ten można opisać za pomocą grafu przejść automatu. Na przykład, każdy regulator reaktora może być włączony albo wyłączony. W stanie wyłączenia regulator wykonuje usługi *Ustaw nastawy* i *Stabilizuj*, a w stanie włączenia – usługę *Wyłącz*. Przejścia między stanami następują w wyniku wykonania usług *Stabilizuj* i *Wyłącz*.



Rys. 4. Model obiektowy programu sterującego reaktorem chemicznym

Najważniejszą cechą modelu obiektowego jest lokalność zarówno danych, jak i przetwarzania ukrytego wewnątrz poszczególnych obiektów. Własność ta utrudnia określenie ograniczeń narzuconych na czynności realizowane przy współpracy wielu obiektów. Specyfikacja takich ograniczeń wymaga określenia scenariusza przetwarzania, tzn. sekwencji wywołań usług, niezbędnych dla realizacji danej czynności i zdefiniowania ograniczeń w stosunku do tego scenariusza. Na przykład scenariusz włączania sterowania reaktorem składa się z następujących wywołań:

Reaktor. Włącz + 3 × (*Pętla regulacyjna. Włącz* + *Regulator. Stabilizuj*)

Jeżeli specyfika rozważanego reaktora wymaga, aby czas włączania nie przekraczał Δt, to ograniczenie to należy odnieść do czasu wykonania przytoczonego scenariusza.

Treść specyfikacji wymagań. Wynikiem analizy obiektowej jest specyfikacja wymagań, w skład której wchodzą:

- model obiektowy, pokazujący relacje i interakcje zachodzące między klasami i obiektami występującymi w dziedzinie zastosowania;
- specyfikacje wszystkich klas i obiektów, obejmujące: definicje atrybutów, definicje usług, diagram stanów i tablicę dostępności usług w różnych stanach;
- opis wymagań niefunkcjonalnych, podany w postaci scenariuszy krytycznych czynności i narzuconych na nie ograniczeń.

Modelowanie współbieżności

Metody oparte na modelowaniu współbieżności dostarczają aparatu pojęciowego i notacyjnego, wystarczającego do formalnego zapisania wymagań i badania pewnych właściwości specyfikacji, nie określają natomiast żadnych użytecznych wskazówek, dotyczących sposobu prowadzenia analizy. Większość metod tej grupy wykorzystuje dwa rodzaje modeli współbieżności:

- procesy współbieżne,
- sieci Petriego.

Dla zilustrowania podstawowych cech tych metod przedstawimy nieco dokładniej postać specyfikacji wymagań, opracowanej w języku PAISLey (*Process-oriented, Applicative, and Interpretable Specification Language*) [9]. Język ten nie osiągnął co prawda nigdy stanu komercyjnej dostępności, jest jednak dobrym przykładem metodologii specyfikacji wykonalnych. Podczas rozwijania języka zostały dla niego opracowane dość bogate narzędzia programowe, umożliwiające statyczną analizę spójności specyfikacji i symulacyjne wykonanie.

Język PAISLey. Specyfikacja wymagań powstaje przez dekompozycję problemu na wiele równoległych *procesów*, odzwierciedlających równoległe działanie środowiska i pracującego w tym środowisku oprogramowania. Każdy proces biegnie w nieskończonej pętli, generując w każdym obiegu nowy *stan procesu*. Każdy stan jest określony przez wartościowanie – czy raczej wiązanie – *zmiennych* tego procesu. Dla wyeliminowania efektów ubocznych, język nie dopuszcza żadnych zmiennych

globalnych – wszystkie zmienne są lokalne wewnątrz poszczególnych procesów.

Definicja procesu składa się z definicji *przestrzeni stanów* i definicji *funkcji przejścia*, której dziedziną i przeciwdziedziną jest przestrzeń stanów. Funkcję przejścia definiuje się w języku funkcjonalnym, określając funkcję przekształcającą bieżący stan procesu w stan następny. Definicja funkcji przejścia może mieć postać wielopoziomowej kompozycji funkcji prostszych, połączonych za pomocą trzech operatorów:

- złożenia funkcji (w matematycznym sensie), czyli sekwencyjnego wykonania funkcji składowych,
- alternatywy, czyli wyboru jednej z kilku funkcji składowych, zależnie od wartości zadanego warunku,
- rozszczepienia, czyli równoległego wykonania funkcji składowych.

Różne procesy, tworzące specyfikację, mogą komunikować się ze sobą wyłącznie za pomocą *funkcji wymiany*. Funkcja wymiany nie jest funkcją w matematycznym znaczeniu tego słowa, lecz stanowi model złożonego kanału komunikacyjnego, wymieniającego synchronicznie wiadomości między współpracującymi procesami. Każda funkcja wymiany ma swoją nazwę (globalną w całej specyfikacji) i atrybut, określający szczegóły jej działania. Ta sama funkcja wymiany może być wywoływana w wielu procesach jednocześnie, a wywołanie jest równoważne zainicjowaniu spotkania z innym procesem. W spotkaniu uczestniczą zawsze dwa procesy, które wymieniają między sobą argumenty wywołania funkcji wymiany; w razie zbiegu wywołań w więcej niż dwóch procesach, współpracująca para jest wybierana niedeterministycznie. Na przykład, wywołanie funkcji wymiany *read* w procesach *P1* i *P2*:

Proces *P1* Proces *P2*

. . . . . .

x-read[*a*] *x-read*[*b*]

zwróci w wyniku wartość *b* w procesie *P1* i wartość *a* w procesie *P2*.

Funkcja wymiany zachowuje więc syntaktykę funkcji matematycznych: wywołana z jakimś argumentem zwraca zawsze jakąś wartość. Nie zachowuje natomiast semantyki funkcji, gdyż zwrócona wartość nie zależy od wartości argumentu. Co więcej, wywołanie funkcji wymiany ma efekt uboczny w postaci wstrzymania obliczenia procesu aż do chwili zakończenia spotkania. Funkcje wymiany są więc narzędziem zarówno komunikacji, jak i synchronizacji procesów.

Przebieg spotkania reguluje atrybut, dopisywany jako przedrostek nazwy w wywołaniu funkcji wymiany. Może nim być:

x – funkcja wywołana z tym atrybutem może wymienić dane z dowolnym wywołaniem tej samej funkcji w innym procesie;

xm – funkcja wywołana z tym atrybutem może wymienić dane tylko z funkcją wywołaną z atrybutem *x* lub *xr;* dwie funkcje wywołane z atrybutem *xm* nie mogą współpracować ze sobą;

xr – funkcja wywołana z tym atrybutem może wymienić dane z dowolnym wywołaniem tej samej funkcji w innym procesie (jak *x*), natomiast nie wstrzymuje biegu procesu; jeżeli żaden inny proces nie jest gotowy do odbycia spotkania, to funkcja wraca natychmiast, zwracając swój argument jako wartość funkcji.

Atrybut *xm* umożliwia modelowanie sytuacji typu jeden producent – wielu konsumentów. Konsumenci pobierają dane od producenta przez ten sam kanał komunikacyjny. Ponieważ konsumenci nie mogą współpracować ze sobą, wywołują więc funkcję wymiany z atrybutem *xm* (*mutual exclusion*). Producent może współpracować ze wszystkimi konsumentami, wywołuje więc funkcję wymiany z atrybutem *x*. Atrybut *xr* (*real-time*) umożliwia modelowanie współpracy procesów, z których jeden musi biec w ustalonym reżimie czasowym, nie dopuszczającym przerw i zatrzymań. Struktura danych przetwarzanych przez procesy jest reprezentowana w specyfikacji w postaci definicji dziedziny i przeciwdziedziny funkcji. Definicje danych są budowane w języku algebry zbiorów, z wykorzystaniem operacji wyliczenia elementów, sumy zbiorów i produktu kartezjańskiego. Wszystkie funkcje są funkcjami jednoargumentowymi, przy czym zarówno argument, jak i wynik funkcji mają zawsze postać listy.

Przykład

Rozważmy ponownie problem sterowania reaktorem chemicznym, ograniczony do zadania rejestracji temperatury w reaktorze. W treści programu uwzględnimy konieczność filtracji i kontroli wartości mierzonej oraz alarmowe opróżnienie reaktora w razie przekroczenia ograniczeń. Specyfikacja w języku PAISLey (rys. 5), składa się z dwóch procesów:

reactor – symulator reaktora,

monitor – monitor temperatury,

z których pierwszy pełni rolę symulatora reaktora, a drugi programu nadzorującego temperaturę.

```
(reactor[initial_state], monitor[initial_value]);
reactor: REACTOR_STATE → REACTOR_STATE;
reactor[x] = proj_1[(simulator[x], dsensor[x])];
simulator: REACTOR_STATE → REACTOR_STATE;
dsensor: REACTOR_STATE → {'null'};
dsensor[x] = xr_sensor[x];
monitor: MONITOR_STATE → MONITOR_STATE;
monitor[v] = proj_1[(filter[(v, msensor[null])], test[v])];
msensor: {'null'} → REACTOR_STATE;
msensor[null] = x_sensor[null];
filter: MONITOR_STATE × REACTOR_STATE → MONITOR_STATE;
test: MONITOR_STATE → {'null'};
test[v] = /out_of_limits[v]: alarm[v], True: null/;
REACTOR_STATE = TEMPERATURE × LEVEL × VALVES;
```

Rys. 5. Specyfikacja złożona z procesów *reactor* i *monitor*

Przestrzeń stanów reaktora tworzy zbiór *REACTOR_STATE*. Funkcją przejścia procesu reactor jest funkcja o tej samej nazwie, której dziedziną i przeciwdziedziną jest zbiór *REACTOR_STATE*. Funkcja *reactor* jest zbudowana z kilku funkcji prostszych, połączonych przez operatory kompozycji. Wewnętrzną część definicji tworzy operator rozszczepienia ( ), który dzieli wykonanie na dwa równoległe wątki wykonania funkcji składowych: *simulator* i *dsensor*. Obliczenie tego fragmentu daje w wyniku dwuelementową listę, zawierającą wyniki obliczenia funkcji składowych. Zewnętrzną część definicji funkcji *reactor* tworzy funkcja *proj-1*, która wybiera pierwszy element listy[1)], i zwraca go jako wartość funkcji *reactor*.

[1)] Tak samo działa operator *car* języka LISP

Element ten określa nowy stan reaktora, stanowiący jednocześnie stan początkowy następnego kroku symulacji. Dekomponując dalej funkcję *simulator* można określić działanie symulatora reaktora z dowolną dokładnością.

Przestrzeń stanów monitora tworzy zbiór *MONITOR_STATE*. Struktura procesu *monitor* jest bardzo podobna do struktury procesu *reactor*. Nowym elementem jest tu tylko operator alternatywy / /, użyty w definicji funkcji *test*. Argumentem operatora jest lista przypadków, której elementami są pary złożone z warunku i funkcji. Wykonanie operatora przebiega w dwóch krokach. W pierwszym kroku oblicza się wszystkie warunki od strony lewej do prawej, aż do znalezienia warunku o wartości *Prawda*. W drugim kroku obliczana jest funkcja odpowiadająca temu warunkowi i obliczona wartość jest zwracana jako wartość całej alternatywy. Warunek *True* ma zawsze wartość *Prawda*.

Obydwa procesy komunikują się przez funkcję wymiany o nazwie *sensor*, wywoływaną w procesie *reactor* z atrybutem *xr*, a w procesie *monitor* z atrybutem *x*. Dzięki takim atrybutom proces symulujący działanie reaktora nie jest nigdy wstrzymywany, lecz biegnie stale z pełną szybkością. Łatwo zauważyć, że użyteczne dane są przekazywane w tym przykładzie w jednym kierunku, od procesu *reactor* do procesu *monitor*.

Uwzględnienie w specyfikacji symulatora instalacji sterowanej jest charakterystyczną cechą wielu metod opartych na wykonalnej specyfikacji – pierwsze próby oprogramowania są wykonywane na symulatorze, który jest traktowany jako integralna część projektu.

Zaletą specyfikacji napisanej w języku PAISLey jest formalnie określona składnia i semantyka oraz wykonalność. Związane z językiem narzędzia programowe umożliwiają automatyczną kontrolę spójności specyfikacji i symulacyjne wykonanie prototypu. Niezwykle wygodną cechą symulatora jest możliwość wykonania niekompletnej specyfikacji, co daje możliwość wczesnej oceny przyjętych założeń.

Wady tej metody są trojakiego rodzaju. Pierwszą jest brak możliwości formalnego wyrażenia wymagań czasowych, które są wprowadzane w formie „znaczących" komentarzy, uwzględnianych przez symulator. Drugą wadą jest brak formalnych narzędzi analizy, zdolnych np. do wykrycia zakleszczeń w specyfikacji – problem ten można jednak rozwiązać przez przetłumaczenie specyfikacji na język sieci Petriego [5]. Trzecim, najlepiej widocznym niedostatkiem języka jest bardzo niska czytelność specyfikacji. Tę ostatnią wadę rekompensuje jednak możliwość obejrzenia efektów wykonania specyfikacji.

* * *

Celem analizy i specyfikacji wymagań jest zrozumienie i udokumentowanie dziedziny zastosowania programów, potrzeb użytkownika oraz sposobu spełnienia tych potrzeb. Prace nad tymi problemami doprowadziły do powstania wielu metod o różnym znaczeniu praktycznym i teoretycznym. W obecnym stanie tych prac największe zastosowanie praktyczne znajdują metody analizy strukturalnej i analizy obiektowej.

Metody analizy strukturalnej dysponują najlepiej poznanym aparatem pojęciowym i notacyjnym i są najpełniej sprawdzone w praktyce przemysłowej. Przejście od modelu analizy do implementacji programu ułatwiają metody projektowania strukturalnego. Wadą tych metod jest pewna trudność w przeniesieniu modelu analizy do etapu projektu szczegółowego oraz niedostateczna modyfikowalność programów opracowanych za pomocą metod projektowania strukturalnego. Metody analizy obiektowej są znacznie mniej poznane i nie zostały jeszcze wystarczająco sprawdzone w dużych projektach przemysłowych.

Naturalną kontynuacją analizy obiektowej są metody projektowania obiektowego, które umożliwiają odwzorowanie modelu analizy w strukturę programu napisanego w języku obiektowym. Przeszkodą w stosowaniu tego podejścia jest niezadowalający stan rozwoju metod projektowania obiektowego i języków programowania, które nie osiągnęły jeszcze poziomu niezbędnego dla zastosowania w przemysłowych systemach czasu rzeczywistego. Nieodłączną cechą takich systemów jest równoległość działania. Dominujący język obiektowy – C++ – jest jednak sekwencyjny, co uniemożliwia bezpośrednie odwzorowanie modelu analizy w strukturę programu. Wydaje się, że w obecnym stanie technologii programowych metody analizy obiektowej znajdują zastosowanie głównie przy specyfikowaniu i projektowaniu programów w języku Ada (który nie jest jednak w pełni obiektowy).

Metody formalne – zarówno oparte na dowodzeniu, jak i na językach wykonalnych – nie oferują jeszcze całościowego rozwiązania problemu analizy i specyfikacji oprogramowania. Tym niemniej, wiele metod formalnych znajduje już cząstkowe zastosowanie przy budowie lub badaniu kluczowych fragmentów specyfikacji wymagań.

LITERATURA

[1] Coad P., Yourdon E.: Object-oriented Analysis. Prentice-Hall, Englewood Clifs, 1991

[2] Hatley D.J., Pirbhai I.A.: Strategies for Real-Time System Specification. Dorset House, 1987

[3] Ross D.T.: Applications and Extensions of SADT, Computer, Vol. 18, No 1, pp. 25–34, 1985

[4] Sacha K.: Projektowanie oprogramowania systemów wbudowanych. Informatyka nr 7, s. 8–15, 1993

[5] Sacha K.: Transformational Implementation of PAISLey Specifications Using Petri Nets, Software Eng. Journal, Vol. 7, No 3, pp. 191–204, 1992

[6] Skidmore S., Farmer R., Mills G.: SSADM Version 4. Models and Methods, NCC Blackwell Ltd., 1992

[7] Wards P.T., Mellor S.J.: Structured Development for Real-Time System Systems, Prentice-Hall, Englewood-Cliffs, 1985

[8] Zave P.: A Comparison of the Major Approaches to Software Specification and Design; in.: Thayer R.H., Dorfman M. (eds.): System and Software Requirements Engineering, IEEE CS Press, pp. 197–199, 1990

[9] Zave P., Schell W.: Salient Fatures of an Executable Specification Language and its Environment. IEEE Trans. Software Eng., Vol. SE-12, No 2, pp. 312–325, 1986

Każdy numer INFORMATYKI
można kupić w redakcji.
Prowadzimy sprzedaż odręczną
i wysyłkową. Wystarczy do nas
zatelefonować lub napisać.

Nasz nowy adres:
00-950 Warszawa, skr. poczt. 1004
ul. Ratuszowa 11, p. 628 i 644
tel., fax: 619-11-61

Windows NT i systemy czasu rzeczywistego (cz. 2)

W poprzedniej części zdefiniowaliśmy pojęcie systemu czasu rzeczywistego i podaliśmy cechy dwóch kategorii tych systemów. W tym odcinku przyjrzymy się jak architektura Windows NT realizuje funkcje systemu czasu rzeczywistego.

Podstawowymi elementami architektury Windows NT, które z pewnością zainteresują projektantów systemów czasu rzeczywistego są:

- Reakcje na zdarzenia zewnętrzne
- Priorytety i zarządzanie wątkami
- Wymagania synchronizacyjne
- Deterministyczny czas reakcji

Reakcja na zdarzenia zewnętrzne . Wymagana jest odpowiedź systemu w określonym przedziale czasu. W Windows NT zarówno zarządzanie przerwaniami jaki i zarządzanie I/O bierze to pod uwagę. Zarówno kernel jak i HAL (Hardware Abstraction Layer) są odpowiednio przystosowane do obsługi tych zadań. Kernel może działać na jednym z 32 możliwych poziomów przerwań, co pozwala na odpowiednie ustawianie priorytetów zadań. Ponieważ Windows NT przeznaczony jest do pracy na systemach wieloprocesorowych, kernel potrafi również rozdzielać obsługę przerwań pomiędzy procesory. Do dyspozycji projektantów systemów czasu rzeczywistego jest również asynchroniczna obsługa I/O. Pozwala to na kolejkowanie I/O i obsługiwanie ich w odpowiednim czasie. Ponadto w Windows NT został zaimplementowany specjalny mechanizm, który powoduje, że obsługa I/O trafia do ostatnio wykonywanego wątku. Zapewnia to, że cache nie został jeszcze naruszony i że system efektywnie wykorzystuje swoją moc. Bardzo ważnym elementem dla systemów czasu rzeczywistego są też sterowniki urządzeń (device drivers). Bardzo istotne jest aby obsługa sterownika urządzenia została zakończona bez żadnych przerwań, czego czasami wymagają aplikacje czasu rzeczywistego.W celu dostarczenia takiego poziomu wydajności kod sterownika urządzenia musi być bardzo dobrze napisany. Ze swej strony Windows NT zapewnia obsługę sterowników całkowicie wewnątrz procesów systemowych i gwarantuje im dostęp do wszystkich urządzeń poprzez HAL.

Priorytety i zarządzanie wątkami. Aplikacje czasu rzeczywistego mają z definicji powiązany ze sobą element czasu. W związku z tym jest ważne aby zrozumieć jak Windows NT przypisuje priorytety do wątków i zarządza ich wykonywaniem. Każda aplikacja zdefiniowana jest jako szereg procesów. Ponieważ Windows NT jest systemem wywłaszczeniowym i wielozadaniowym, pozwala to na jednoczesne wykonywanie wielu aplikacji w tym samym czasie. Każdy proces ma powiązane ze sobą właściwości, które go charakteryzują. Dla systemów czasu rzeczywistego jedną z najważniejszych właściwości jest priority class (taka jak real_time), która definiuje podstawowy priorytet, z którym aplikacja będzie uruchamiana. Równie ważnie jest zarządznie pamięcią. Windows NT rozwiązuje wiele problemów, z którymi projektanci systemów czasu rzeczywistego spotykają sie w innych systemach. Przede wszystkim stronicowanie I/O odbywa się z priorytetem niższym niż priorytet zadań czasu rzeczywistego. Po drugie NT pozwala aplikacji na "zamknięcie " się w pamięci, co zapobiega stronicowaniu wewnątrz własnych procesów aplikacji. Windows NT pozwala również na mapowanie pamięci. Dzięki temu różnorodne procesy, sterowniki urządzeń i aplikacje użytkownika mogą współdzielić tą samą pamięć fizyczną. W rezultacie uzyskuje się bardzo szybką komunikację pomiędzy współpracującymi wątkami lub pomiędzy sterownikiem i aplikacją. Kolejnym elementem istotnym dla systemów czasu rzeczywistego jest zarządzanie buforem (cachem). Ma on pewnie wpływ na nieprzewidywalność czasu reakcji systemu na zdarzenie, ponieważ w systemach operacyjnych ogólnego zastosowania, takich jak Windows NT, nie sposób jest prezycyjnie określić, czy dana porcja danych znajdzie się w cache'u, czy też nie.

Dokończenie w następnym odcinku.

0/1/95

informacje

W dniach 22–25 listopada 1995 r. odbędzie się w Krakowie już po raz trzeci impreza informatyczno-telekomunikacyjna pod nazwą INFO-FESTIWAL '95, organizowana przez Centrum Promocji Informatyki, pod patronatem merytorycznym Komitetu Badań Naukowych, Pełnomocnika Premiera ds. Informatyki i Polskiej Izby Informatyki i Telekomunikacji. Redakcja nasza objęła patronat prasowo-promocyjny nad tym festiwalem.

Wystawie, zorganizowanej w trakcie INFO-FESTIWALU będą towarzyszyć konferencje, seminaria, prezentacje. Odbędzie się premiera polskiej wersji programu Windows '95. Planowany jest udział ponad 400 przedstawicieli instytucji i jednostek naukowych z całej Polski. Po raz pierwszy została ufundowana nagroda przewodniczącego KBN za najciekawsze prezentowane rozwiązanie teleinformatyczne dla potrzeb nauki.

Ekspozycje zostały zrealizowane w pięciu punktach Krakowa, są to:
- ◇ Galeria BWA,
- ◇ Muzeum Narodowe,
- ◇ Pałac TPSP,
- ◇ Filharmonia Krakowska,
- ◇ Japońskie Centrum Sztuki i Techniki.

Koszt udziału w wystawie wynosi równowartość 120 USD (plus VAT) za 1 m^2 zabudowanego stoiska w systemie OCTA-NORMA. Przewidywane są rabaty po 5%: dla członków PIIiT, firm uczestniczących poprzednio w INFO-FESTIWALU oraz tych, które zgłosiły swój udział do końca czerwca br. 10% rabatu udziela się firmom za stoisko o powierzchni powyżej 15 m^2.

Koszty uczestniczenia w konferencjach i seminariach są różne, od 100 zł do 360 zł, szczegóły podają organizatorzy.

Wystawa będzie czynna od 22 do 25 listopada w godz. 10.00–18.00. Wstęp dla zwiedzających – 5 zł, dla uczestników konferencji – bezpłatny. Organizatorzy zapewniają bezpłatny transport mikrobusami pomiędzy ekspozycjami.

Kalendarz imprez

21–23.XI.95
Zaawansowane technologie informatyczne w nauce

21–22.XI.95
Informatyka w mediach

22.XI.95
Panel szefów największych firm teleinformatycznych

23.XI.95
Poufność i bezpieczeństwo danych w komputerowych bazach danych

24–25.XI.95
Internet w Polsce

24–25.XI.95
Teleinformatyka w wojsku i policji

23–24.XI.95
Systemy informacji przestrzennej

Imprezą inaugurującą INFO-FESTIWAL '95 będzie Panelowe Forum Dyskusyjne pod nazwą CENA SUKCESU – WARUNKI ROZWOJU z udziałem szefów największych polskich firm teleinformatycznych, przedstawicieli administracji rządowej, organizacji informatycznych, autorytetów naukowych i prasy.

Ekspozycje tematyczne

Informatyka dla nauki

Internet w Polsce

Systemy informacji przestrzennej

Teleinformatyka w wojsku i policji

Poczty i sieci teleinformatyczne

Informatyka dla biznesu

Multimedia

Forum odbędzie się 22 listopada. Przebieg dyskusji zostanie zarejestrowany przez TV-Kraków i będzie wyemitowany w formie reportażu następnego dnia.

W ramkach zamieszczamy ogólne informacje o ekspozycjach, konferencjach i seminariach, szczegóły podamy w numerze listopadowym.

30 listopada i 1 grudnia odbędzie się także w Krakowie konferencja **„Teleinformatyka w turystyce i hotelarstwie"**.

Jest to oficjalna impreza towarzysząca Targom HORECA MID EUROPE. Koszt uczestnictwa jednej osoby wynosi 450 zł.

Wszystkie dodatkowe informacje można otrzymać telefonicznie:

Centrum Promocji Informatyki
00-503 Warszawa 15, skr. poczt. 44
ul. Żurawia 4a
Tel.: (0.22) 621.76.26, 693.59.46, 693.59.22
Faks: (0.22) 693.59.49, 693.59.58

Biuro Organizacyjne
INFO-FESTIWAL '95 w Krakowie
Fundacja Monumentis Patriae
ul. Piłsudskiego 12/14
31-110 Kraków
Tel.: (0.12) 22.27.63 w. 17
Tel., faks: (0.12) 22.27.33 w. 17

Nowy Deskpro XL, nowa ProLinea

Compaq z Pentium – dobrana para

Zegar 120 MHz, niebawem 133 MHz standardowo 16 MB RAM

Compaq rozszerza serię Deskpro XL o nowy model stacji roboczej z procesorem Pentium 120 MHz. Wprowadza też nowe modele serii ProLinea i Deskpro: Compaq ProLinea 5120 i Compaq Deskpro 5120 standardowo są wyposażone w Pentium 120 MHz. Maszyny z procesorem Pentium 120 MHz są wyposażone standardowo w 16 MB pamięci RAM, dysk twardy o pojemności 1 GB, architekturę Tri-Flex/PCI, magistralę lokalną PCI i kartę graficzną. Linia Deskpro XL ma też inne rozszerzenia technologiczne, m.in. Silent-Cool Fan, który redukuje hałas i zwiększa wydajność chłodzenia wszystkich części PC i Compaq Vocalyst Keyboard – klawiaturę, w którą są wbudowane głośniki i mikrofon. Wkrótce Compaq zamierza stosować w stacjach roboczych i serwerach nowy procesor Pentium 133 MHz.

Coraz łatwiej pracować

Biuro w terenie

Weź ze sobą notebooka Zbierz dane i przekaż je do centrali

Firma Intergraph Software Solutions oferuje nowy pakiet programowy – Field Office. Jest on przeznaczony dla pracowników technicznych, inspektorów terenowych i innych osób, które podczas pracy w terenie mogą używać przenośnych komputerów, tak jakby znajdowały się w biurze, korzystając przy tym z zasobów w centrali, porównując i przekazując dane.

Field Office składa się z trzech części. FieldWorks oferuje proste i wydajne procedury służące integracji danych zawartych w tzw. elektronicznych książkach polowych (EFB) ze środowiskiem CAD. **CogoWorks** – to zaawansowany moduł tworzenia geometrii, np. układu ulic, działek, budynków. Zawiera m.in. aplikację DraftWorks, służącą do tworzenia raportów liczbowych. Trzecia część – **MikroStation Review** to łatwy w użyciu program do szybkiego przeglądania plików graficznych. Służy m.in. do porównywania stanu budowy z projektem.

Program Edukacyjny IBM

OS/2 Warp za 39 zł

Każda instytucja edukacyjna w Polsce, która zdecyduje się wziąć udział w ogłoszonym przez IBM Programie Edukacyjnym, będzie mogła zainstalować na swoim komputerze za jedyne 39 zł system operacyjny OS/2 Warp. Tyleż samo zapłaci za każde dodatkowe oprogramowanie pracujące pod OS/2 Warp, m.in. PC DOS v. 7, LAN Serwer 4.0, C SET++, bazy danych, programy multimedialne i inne. Szczegółowe informacje na temat Programu Edukacyjnego można otrzymać u partnerów handlowych IBM Polska, realizujących ten program: w firmie PERYT (tel/faks [022] 411707) i TCH Systems (tel. [022] 487172, faks 481206).

Nokia 2110

Nowy przenośny telefon

Czeka na połączenie Dostęp z komputera

Nokia Mobile Phones wprowadziła na rynek przenośny telefon – Nokia 2110, który działa w sieciach cyfrowych GSM, może więc być używany także do wysyłania i odbioru danych i krótkich wiadomości. W porównaniu z poprzednimi modelami ma on nowe funkcje, takie jak oczekiwanie na połączenie i trzymanie go, oraz dostęp głosowo-pocztowy z komputera typu *desktop*. Nowe urządzenia do ładowania, dostępne dla telefonu Nokia 2110, mogą dłużej utrzymać żywotność baterii dzięki użyciu przycisku zasilającego na podstawie *desktopa*. Nokia 2110 zawiera oryginalną kartę danych standardowego interfejsu PCMCIA – Nokia PC Card.

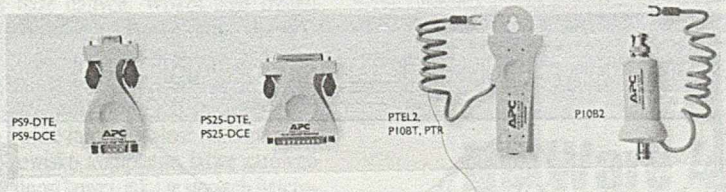
Nowa Nokia Cellular Data Card (komórkowa karta danych) dla GSM jest wprowadzana do komputera klasy PC i łączona z przenośnym telefonem za pomocą prostego przewodu. Serwis krótkich wiadomości może być również prowadzony z *notebooka*.

ProtectNet

Ochrona przed przepięciami

Cała rodzina urządzeń Chroni wszystko

ProtectNet to nowa rodzina urządzeń firmy American Power Conversion przeznaczona do pełnej ochrony sieci LAN, komputerów klasy PC oraz urządzeń elektronicznych powszechnego użytku. Są to samodzielne moduły, które umożliwiają ochronę przed przepięciami dla kart sieciowych, portów szeregowych, linii telefonicznych (faks, modem). Rodzina ProtectNet jest dostępna w trzech kategoriach, umożliwiając ochronę portów sieciowych, szeregowych, linii telefoniczno-modemowych. ProtectNet można zainstalować na przewodzie łącza danych, doprowadzonych bezpośrednio do komputera.



3Com oferuje

Nowy przełącznik z rodziny SuperStack

Technologia *plug-and-play* Odporność na błędy

3Com oferuje nowy tani przełącznik dla sieci Ethernet – LinkSwitch 500, należący do rodziny SuperStack. Jest to przełącznik dla grup roboczych, wykonany w technologii *plug-and-play*, z 10 portami w standardzie Ethernet, o dużych możliwościach dostosowawczych. Charakteryzuje go również łatwość skonfigurowania. Przełączenia realizowane przez LinkSwitch wykorzystują technologię układów dedykowanych ASIC (*Application-specific Integrated Circuit*). LinkSwitch może być sterowany poleceniami zgodnymi z protokołem SNMP, jak również przez aplikacje do zarządzania siecią Transced produkcji 3Com. LanSwitch umożliwia przełączanie z prędkością przewodową 10 Mbps na wszystkich 10 portach. Wszystkie produkty 3Com należące do systemu SuperStack są odporne na błędy.

Opracowała KRYSTYNA KARWICKA-RYCHLEWICZ

ZACZĘŁA SIĘ ERA WINDOWS '95

Wprawdzie „Informatyka" nie jest pismem, którego głównym obszarem zainteresowań byłyby komputery osobiste i ich zastosowania w życiu codziennym, to jednak nie możemy pominąć milczeniem pojawienia się na rynku systemu „Windows '95" – nawet biorąc pod uwagę, że jego premiera rynkowa odbyła się pod koniec sierpnia i że w żaden sposób nie jesteśmy w stanie konkurować szybkością informacji z prasą codzienną.

Nie możemy nie wspomnieć o „Windows '95", jest to bowiem program zbyt ważny, zmieniający – naszym zdaniem – świat komputerów osobistych bardziej, niż to zrobiło Pentium. Używając ostatecznej wersji testowej „Okien '95" przez kilka tygodni, nie mam wątpliwości, że jest to najłatwiejszy i najdoskonalszy w użyciu system operacyjny w historii informatyki. Szczególną cechą tego systemu jest technika „Plug and Play". Dlatego nie mam Microsoft-owi za złe ani opóźnienia (w końcu, system jest programem zbyt ważnym, by go wypuścić na rynek bez odpowiedniego „odpluskwienia", ani nie gorszę się z powodu ogromnego hałasu reklamowego wokół sprawy.

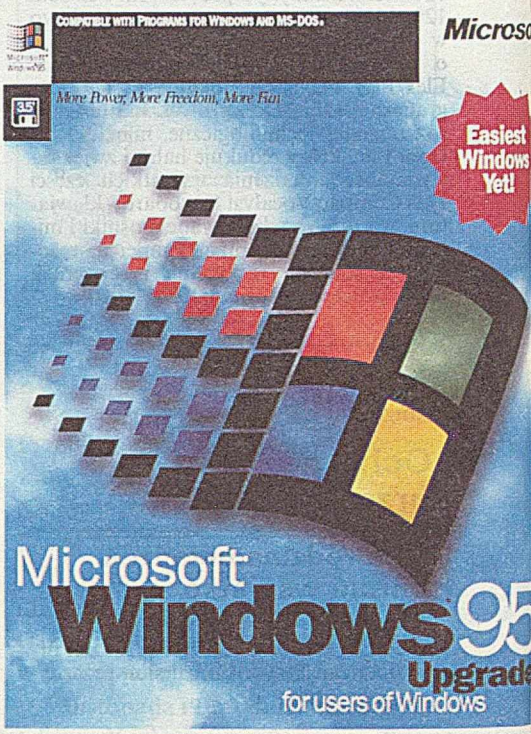
Nawiasem mówiąc – żeby nie było podejrzeń, że mój nie skrywany entuzjazm jest niecną kryptoreklamą: oto kilka tajemniczych wydarzeń, które odnotowałem zainstalowawszy wspomnianą wersję tekstową wersji „Paneuropean" systemu na komputerze „Optimus" Pentium 60 z 8 MB RAM jako *upgrade* na *„Windows for Workgroups 3.11 PL"*. Po pierwsze, *„zginęły"* mi natychmiast niektóre polskie kroje pism (ale częściowo: Word 6 ich nie dostrzegał, ale WordPad i Notatnik – i owszem), które zostały zastąpione krojami skrajnie uproszczonymi – tak było np. z krojem *Times New Roman CE;* co jeszcze ciekawsze, po tygodniu wszystko samoczynnie wróciło do normy. Po drugie, po zainstalowaniu i późniejszym usunięciu (nie chciały się uruchomić) *bety* Norton Utilities for Win '95 komputer zaczął się domagać, bym usunął wszelkie odwołania do pewnego pliku w pliku SYSTEM.INI, w którym – jako żywo – żadnych takich odwołań nie ma (w niczym to zresztą nie przeszkadza, tyle że komputer się „awanturuje" bez sensu). Po trzecie, prawidłowo zainstalowany napęd CD-ROM przestał (ale nie natychmiast: po pewnym czasie) być widziany przez system; jest natomiast widziany wtedy, gdy uruchomi się spod Windows '95 sesję DOS-u ...

Piszę o tym dla ciekawości raczej; nie krytykuję systemu, bo nie uruchomiłem jeszcze jego wersji handlowej, a w końcu testy są po to właśnie, by owa wersja handlowa była dobra. Swoją drogą, gdyby mi ktoś umiał to wszystko wytłumaczyć – stawiam kawę.

W każdym razie: w dwudziestolecie swego istnienia Microsoft pokazał kawał dobrej roboty. Cieszy też, że Polska wyraźnie awansowała w hierarchii zainteresowań Amerykanów: o ile poprzednio czekaliśmy na w pełni spolszczoną wersję systemu około roku, to teraz została zapowiedziana na 24 października, zaś na dobrą sprawę można używać bez problemów jeszcze wcześniejszej wersji paneuropejskiej. Podobnie ma być z programami użytkowymi.

Odnotowując wszystko to z satysfakcją pragniemy naszym Czytelnikom pokazać – no, skoro nie możemy więcej, to choćby pudełka „Windows '95" i czegoś, co jednak okazało się niespodzianką: dostosowanego do systemu pakietu narzędziowego „Plus!".

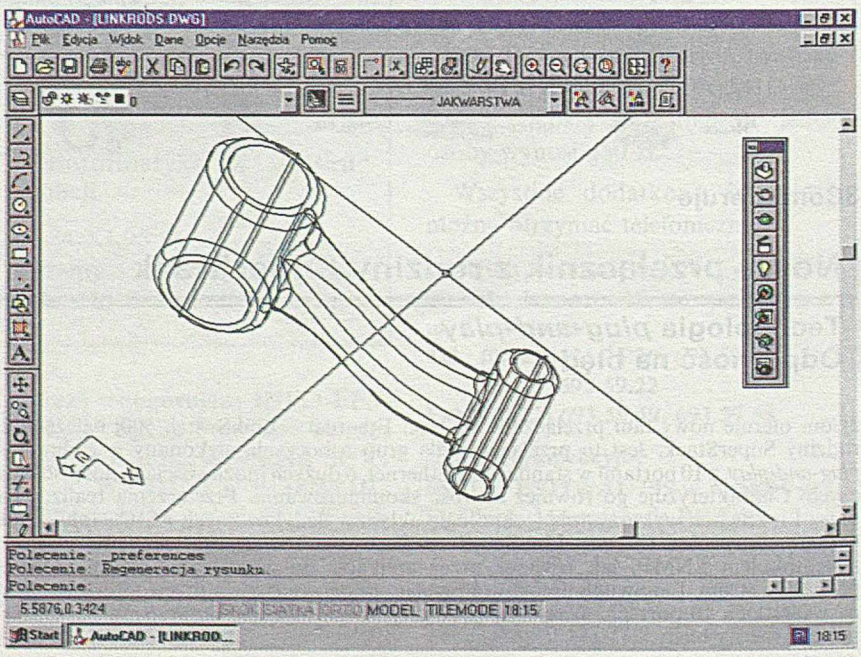
Bogdan MIŚ



Autodesk

AutoCAD rel. 13 po polsku

AutoCAD Release 13 firmy Autodesk – popularne narzędzie wspomagające projektowanie – w niespełna rok po światowej premierze doczekało się polskiej wersji. Została ona opracowana przez firmę CAD Expert z Łodzi. Tak jak i w wypadku wersji angielskiej, licencja na program obejmuje systemy: MS-DOS, Windows, Windows NT. Wraz z polską wersją użytkownik otrzymuje polskojęzyczną dokumentację.



W AutoCAD Release 13 znajdziemy wiele ulepszeń, które ułatwiają i przyspieszają łatwiejsze wykonywanie rysunków i modeli 3D wysokiej jakości. Wyposażono go w nowy elastyczny interfejs Windows, który znacznie podnosi efektywność pracy z narzędziem, a ponadto sprawia, że jest ono łatwiejsze do nauczenia. Przyciski, aktywujące funkcje systemu, są obecnie zgrupowane w paskach narzędzi, których położenie na ekranie może być zmieniane. System udziela użytkownikowi wskazówek dotyczących znaczenia przycisków. Dzięki zdolności dołączania i osadzania obiektów, AutoCAD może łatwo wymieniać dane, takie jak tabele, rysunki, animacje itp., z innymi aplikacjami. Istniejące aplikacje mogą być w prosty sposób modyfikowane. Wersja 13 umożliwia odczytywanie plików w formatach poprzednich wersji. Dodatkowo polecenie *Save as* umożliwia zapisywanie danych w formatach wersji 11 i 12, a także w formacie AutoCAD LT. Po raz pierwszy dokumentacja jest publikowana na dysku CD-ROM.

Najnowszą wersję programu Autodesk wyposażył w nowy sterownik graficzny WHIP, który umożliwił dodanie do AutoCAD Release 13 dwóch całkowicie nowych funkcji: wyświetlania dużych obrazów w czasie rzeczywistym i zmiany ogniskowej. W pakiecie udoskonalono też funkcję wymiarowania (ang. *dimensioning*), redagowania wyglądu tekstu, czcionki i rozszerzania pakietu o dodatkowe opcje i pliki. Nowy sterownik zwiększa sprawność pakietu używanego na platformach Windows 3.1, Windows NT, a także Windows '95. Wraz z programem jest dostarczany moduł konwersji geometrii ze standardu AME (*Advanced Modeling Extension*) do postaci ACIS – standardu, który pozwala tworzyć i modyfikować obiekty za pomocą prymitywów, swobodnie modyfikowanych krzywych i powierzchni oraz operacji logicznych na obiektach podstawowych. Użytkownicy AutoCAD-u mogą otrzymać bezpłatnie nowe funkcje i pliki zawierające udoskonalenia za pośrednictwem CompuServe (GO ACAD), jak również za pośrednictwem dealerów i całej sieci handlowej Autodesku. **(kar)**

Konferencja ACI

Apple nie boi się Windows '95

Środowisko relacyjnych baz danych rodziny 4th Dimension (4D) firmy ACI (*Analyse Conseils Informations*) było tematem konferencji promocyjnej, która odbyła się we wrześniu br. w Warszawie. Impreza przygotowana została wspólnie z firmą Apple, dla której to platformy systemowej produkt ACI pierwotnie został stworzony i w której odnosi największe sukcesy. Zaprezentowano również wersję 4D dla środowiska Windows.

W zależności od potrzeb rodzina 4D oferuje jednostanowiskowy system zarządzania bazą danych z możliwością projektowania samodzielnych aplikacji, 4D Server – podobny zestaw, ale o architekturze klient-serwer, mogący obsłużyć do 125 użytkowników, oraz 4D First – popularną, osobistą bazę danych. Wokół tego trzonu powstały dodatkowe grupy programów usprawniające pracę. Aplikacje pomocnicze obejmują kompilator aplikacji, narzędzia edytorskie, moduły współpracy z innymi bazami danych, system archiwizacji i wiele innych. Całość reklamowana jest jako środowisko kompletne, łatwe zarówno w instalacji jednostkowej, jak i w sieciach heterogenicznych, intuicyjne w obsłudze i programowaniu. Obecnie istnieją wersje dla MacOS i Windows, docelowo 4D zaś ma być w pełni wieloplatformowy.

Przedstawiciele Apple Computers omówili swoją ofertę sprzętową, która obejmuje całą gamę urządzeń – od komputerów osobistych, przez serwery do urządzeń peryferyjnych. Stwierdzili też, iż nie obawiają się ewentualnego spadku zainteresowania produktami Apple jako platformy dla 4D, mimo powstania wersji 4D dla środowiska Windows 95. Stałym atutem jest według nich bardzo dobry sprzęt oparty na PowerPC, powszechnie uznany system operacyjny MacOS, niskie ceny szkoleń oraz korzystne warunki zarówno dla pośredników, jak i zwykłych nabywców. **(P.B.)**

Pięć programów podstawowych

CorelDRAW 6 zgodny z Windows '95

Tuż przed warszawską premierą Windows 95 został zaprezentowany polskim użytkownikom CorelDRAW 6. Jest to 32-bitowa aplikacja, zgodna z Windows 95 i z Windows NT. Pakiet składa się z pięciu modułów: Corel Draw – edytor grafiki wektorowej, Photo – Paint – edytor obrazów rastrowych, Presents – do przygotowywania prezentacji, Dream 3D – do tworzenia trójwymiarowych modeli oraz Motion 3D – do tworzenia prostych animacji. Te dwa ostatnie moduły są nowością. Motion 3D jest uzupełniony przez cztery dodatkowe programy: Corel Depth (program umożliwiający uzyskiwanie trójwymiarowych efektów), Trace (wzbogacony o funkcje OCR, umożliwiający automatyczne rozpoznawanie pisma na dokumentach wprowadzanych do komputera za pomocą skanera), Capture (kopiowanie fragmentów ekranu) oraz Font Master (ułatwiający zarządzanie czcionkami i zawierający bogate biblioteki fontów, wykresów, tekstur itp.). Wszystkie programy wchodzące w skład CorelDRAW 6 są wzbogacone o nowe funkcje. Użytkownik może, na przykład, sam dobierać liczbę ikon w poszczególnych narzędziach pojawiających się na ekranie i dowolnie je ustawiać. Przygotowane rysunki można wysłać faksem lub pocztą elektroniczną bezpośrednio z CorelDRAW lub z dowolnej aplikacji. CorelDRAW 6 wymaga co najmniej komputera 486DX z 8 MB pamięci RAM, ale bardziej polecany jest komputer z Pentium i 16 MB pamięci RAM. I oczywiście system operacyjny Windows 95 lub Windows NT. Pełna wersja programu kosztuje 1690 zł + VAT, upgrade do wersji 5.0 – 1080 zł + VAT, zaś upgrade do wersji 3 i 4 – 725 zł + VAT.

(kar)

NetWare 4.1 o 20% tańsze

Do końca października można kupić taniej o 20% uaktualnione oprogramowanie sieciowe do wersji NetWare 4.1 firmy Novell. Promocja systemu NetWare 4.1 prowadzona jest równolegle z promocją pakietu Novell GroupWise, w ramach której jest oferowany również 20-procentowy upust przy uaktualnianiu oprogramowania GroupWise dowolnego typu, a także przy przechodzeniu od dowolnego konkurencyjnego pakietu obsługi poczty elektronicznej. System NetWare 4 jest zainstalowany już na około sześciu milionach stanowisk i według firmy IDC, która specjalizuje się w analizie rynków, oprogramowanie NetWare 4 jest drugim (po NetWare 3 Novella) najlepiej sprzedającym się sieciowym systemem operacyjnym. Zapewnia on wszystkie siedem podstawowych funkcji sieciowych (obsługę plików, katalogów, drukowanie, zabezpieczanie danych, informowanie za pomocą komunikatów, wieloprotokołowe kierowanie danymi i zarządzanie nimi). Firma Novell oferuje też klientom specjalne szkolenie w zakresie systemu NetWare 4.1 na jednodniowych kursach First Class prowadzonych na całym świecie w autoryzowanych centrach edukacyjnych Novella. Od lipca do połowy października można było wziąć udział w organizowanych co dwa tygodnie w sieci CompuServe konferencjach, w których uczestniczyli członkowie kierownictwa firmy Novell. Informacje na temat uaktualnienia oprogramowania sieciowego są dostępne też w serwisie World Wide Web (http://www.netware.com). **(k)**

MICROSOFT EUROPA I GALlup przeprowadziły ostatnio badania opinii 2000 Europejczyków z W. Brytanii, Francji i Niemiec na temat ich poglądów, zachowań i odczuć związanych z komputerami i techniką informacji w pracy, domu i szkole. Wynika z nich, że ponad połowa Europejczyków uważa, iż dzięki komputerom praca stała się przyjemniejsza. Dwie osoby na pięć uważają, że komputer PC jest niezbędny w domu, a jedna osoba na dziesięć zamierza kupić komputer w tym roku. We Francji 22% użytkowników płci męskiej woli spędzać dodatkową wolną godzinę w ciągu dnia przy komputerze, niż z przyjaciółką. W Wielkiej Brytanii 75% badanych pracę z komputerem określa jako „frajdę", natomiast w Niemczech tylko 22%.

★

EMERSON COMPUTER POwer oferuje nowy zasilacz jednofazowy o mocy 10 kVA. Wzbogacił on serię AP 200, w skład której do tej pory wchodziły jednostki o mocy 3 i 6 kVA. Wykorzystują one oprogramowanie PowerMon współpracujące z większością systemów operacyjnych. Mogą być także wyposażone w adapter SNMP. Dystrybutorem zasilaczy Emersona jest firma Camco z Warszawy.

★

W OFERCIE ABC DATA ZNAlazły się produkty dla systemów sieciowych LAN firmy Allied Telesyn (uprzednio Allied Telesis). Głównymi produktami są wieloportowe *transceivery* inaczej: pełny duplex) o różnych konfiguracjach, wieloportowe *mikrorepeatery* („donosiciele"), rozgałęziacze (*huby*) UTP (od bardzo małych i prostych do dużych i wyposażonych w moduły zarządzające SNMP), bridge, koncentratory światłowodowe, BNC i UTP. Jednym z najnowszych produktów Allied Telesyn jest system AT-3600, który został zaprojektowany przy wykorzystaniu piętrowych rozgałęziaczy do pracy grupowej, przy wykorzystaniu technologii Ethernet. Elementy systemu mogą być konfigurowane w dowolny sposób, umożliwiający optymalną pracę zarówno w małych biurach, jak i w dużych ośrodkach.

★

FIRMA MICROSOFT POLSKA, która 30 czerwca zakończyła swój kolejny rok finansowy (działa od 1992 r.), poinformowała, że sprzedaż jej usług i produktów zwiększyła się o 80% w stosunku do roku ubiegłego. Największy przyrost sprzedaży odnotowano w grupie produktów serwerowych – BackOffice, obejmującej Windows NT Serwer, SQL Serwer, Systems Management Server.

★

AUTODESK INC. POINFORmował o posiadaniu własnego adresu w strukturze WWW w Internecie. Adres dostępny za pośrednictwem Web Browser brzmi: **http://www. autodesk.com.** Zawarte na serwerze informacje uszeregowane zostały w 12 kategoriach.

★

DIGITAL EQUIPMENT CORporation zaprezentował dwa nowe systemy. Jednostką centralną w modelach 5/300 i 5/266 Alpha Station jest RISC-owy mikroprocesor Alpha 21164, pierwszy z mikroprocesorów wytwarzanych seryjnie, który może wykonać miliard operacji na sekundę (1200 MIPS-ów). Ta ogromna moc jest wspomagana przez nowo opracowaną architekturę systemu, zoptymalizowaną pod kątem maksymalizacji wydajności aplikacji używanych na stacjach roboczych. AlphaStation 600 5/300 jest pierwszym komputerem jednoprocesorowym, który pokonał bariery 300 SPECint92 oraz pierwszym, który osiągnął 639.6 Jobs/minutę (zadań/min) według pomiarów AIM.

★

COMPAQ I MICROSOFT PODpisały umowę OEM dotyczącą pakietu Microsoft BackOffice – zintegrowanej rodziny oprogramowania serwerowego. Jeszcze w tym roku Compaq wprowadzi na rynek nowe wersje serwerów, które będą dostarczane z całą rodziną aplikacji serwerowych Microsoft BackOffice dla systemu operacyjnego Windows NT. Klienci mogą dzięki tej umowie spodziewać się obniżonych kosztów eksploatacji, lepszego zarządzania systemami, opracowywanych wspólnie programów z zakresu serwisu i obsługi klienta. Umowa obejmuje obsługę techniczną, pakiet SmartStart i Business Solutions Partnership.

★

NOVELL INFORMUJE, IŻ GroupWise dwukrotnie zwiększył w 1994 r. swój udział na rynku sieciowych systemów przekazywania wiadomości. GroupWise był pierwszym – i, jak dotychczas, jedynym – systemem przekazywania wiadomości, integrującym w sobie jednocześnie pocztę elektroniczną, kalendarz, faks, wymianę informacji w sieci Internet, zarządzanie zadaniami. Listy dostępu (*Accesss Lista*) umożliwiają dostęp innym użytkownikom do wiadomości GroupWise, a także funkcje wysyłania i odbierania informacji bezpośrednio ze skrzynki pocztowej.

★

NOVELL I MICROSOFT PODpisały umowę o wzajemnej współpracy. Jej kluczowymi elementami są wspólne przedsięwzięcia, dotyczące autoryzowanej pomocy technicznej i organizowanie wspólnych działań, mających na celu m.in. podejmowanie dyskusji na temat wymiany oprogramowania, informacji technicznej oraz organizowania programów szkoleniowych. Umowa była wspólną inicjatywą szefów obu firm: Boba Frankenberga (Nowell) i Billa Gatesa (Microsoft). Jedną z najważniejszych korzyści dla klientów jest zestaw informacji technicznych, których współautorami są eksperci techniczni obu firm. Novell i Microsoft już obecnie dzielą się dostępem do produktów, materiałów pomocy technicznej, materiałów szkoleniowych i zasobów technicznych.

★

DELL POLSKA WPROWADZIŁ bezpłatne połączenia telefoniczne (*tollfree*) dla użytkowników swoich komputerów: **0-800 20011** – dział techniczny, **0-800 20012** – dział sprzedaży. Według statystyk 80–90% problemów zgłaszanych przez użytkowników można rozwiązać rozmawiając z fachowcem przez telefon.

★

TEGOROCZNYM ZWYCIĘZCĄ rankingu prowadzonego przez amerykański miesięcznik *LAN The Network Solution Magazine* został ComputerLand/VANSTAR Inc., który uzyskał przychód ze sprzedaży instalacji sieci lokalnych w wysokości ponad 750 mln USD. ComputerLand Poland jest jednym z czołowych przedstawicieli tego międzynarodowego koncernu.

W WARSZAWSKIEJ SZKOLE Głównej Handlowej uruchomiono pierwszą w Polsce instalację sieci standardu ATM wykorzystującą metodę LAN Emulation. Sieć ta obejmuje ponad 1000 stanowisk, a jej struktura opiera się na dwóch przełącznikach (*switchach*) dostępowych ForeRunner LAX-20 firmy FORE Systems. Znajdują się one w dwóch budynkach i są połączone ze sobą linią ATM STM-1 (OC3), pracującą z prędkością 155 Mb/s. Do jednego przełącznika jest dołączonych 16 podsieci Ethernet, do drugiego – 4. Wykorzystując metodę LAN Emulation można na tym szkielecie tworzyć sieci wirtualne.

★

FIRMA 3COM, OFERUJĄCA zaawansowane urządzenia sieciowe, ogłosiła rezultaty finansowe w roku finansowym 1995, który zakończył się 31 maja 1995 r. Sprzedaż wyniosła 1,3 mld USD i wzrosła o 57% w stosunku do 1994 r., zyski wyniosły 125,7 mln USD. Największą sprzedaż odnotowano w okresie od marca do maja br. Największe obroty odnotowano w Niemczech; oddział 3Com w Monachium jest odpowiedzialny za działalność firmy na terenie Europy Centralnej i Wschodniej. Niedawno 3Com otworzył swoje biuro w Warszawie.

★

INTERNATIOAL DATA CORporation (IDC) opublikowała listę największych światowych firm zajmujących się edukacją z zakresu informatyki. Listę sporządzono na podstawie dochodów z działalności edukacyjnej i szkoleń w 1994 r. Na czele listy jest IBM (300 mld USD), następnie HP (165 mln USD), Oracle (122 mln USD), Digital (100 mln USD), Excutrain (95 mln USD), CAP Gemini (82 mln USD), Sun Microsystems (80 mln USD).

★

INTEL OGŁOSIŁ, IŻ W DRUgim kwartale 1995 r. dochody firmy wzrosły o 41% w stosunku do analogicznego okresu 1994 r. i osiągnęły wartość 3,89 mld USD, o 36% wzrosły zyski na akcję. Natomiast zysk netto wzrósł z 640 mln USD (w II kwartale 1994 r.) do 879 mln USD. Firma podniosła swoje planowane wydatki kapitałowe do ogólnej sumy 3,5 mld USD inwestując ok. 300 mln USD. W całym I półroczu br. przychody firmy wzrosły w porównaniu z analogicznym okresem ub. r. o 37% do poziomu 7,45 mld USD, zaś zysk z 1,26 mld USD do 1,77 mld USD.

★

DRUGI KWARTAŁ BR. BYŁ bardzo korzystny dla IBM: zysk wyniósł 1,7 mld USD, co w przeliczeniu na jeden udział wynosi 2,97 USD (w analogicznym kwartale ub. r. – 1,14 USD). Przychody IBM wzrosły w tym czasie o 14,2% (w porównaniu z analogicznym okresem ub. r.) – do 17,5 mld USD. Największy wzrost przychodów zanotowano w Azji i rejonie Pacyfiku (wzrost o 30,9%), o 19,7% wzrosły dochody IBM w Ameryce Łacińskiej, w Europie, na Środkowym Wschodzie i w Afryce przyrost wyniósł 11,5%. Najniższy wzrost odnotowano w Ameryce Północnej – 8,9%, ale i tak IBM osiąga tam największe dochody – 7,1 mld USD.

Opracowała:
Krystyna Karwicka-Rychlewicz

Druga młodość

Quattro Pro 5.5

- **Jak w Windows,**
- **DOS jednak żyje**

Wszyscy ci, którzy już skazali arkusz kalkulacyjny Quattro Pro na zamknięcie w skansenie – srodze się rozczarowali. Novell pod koniec czerwca br. wprowadził na rynek Quattro Pro 5.5 for DOS, czyli najnowszą wersję tego doskonałego arkusza kalkulacyjnego. Quattro Pro jest – jak oświadczyła Dorothy Wise z Novell Business Applications – kamieniem milowym w rozwoju arkuszy kalkulacyjnych i nadal ma ogromną rzeszę użytkowników.

Wersja 5.5 ma wbudowane mechanizmy, pozwalające na łatwą współpracę z NetWare, jest też w pełni zintegrowana z WordPerfectem do DOS-u. Arkusz ma 50 nowych *Quick Templates* (mechanizmy, automatyzujące wykonywanie typowych czynności), które ułatwiają przygotowanie faktur, zamówień, zestaw wydatków i wiele innych.

Użytkownicy mogą dopasowywać szablony do swoich wymagań lub opracowywać swoje własne interfejsy użytkownika. Funkcja *SpeedFill* pozwala na inteligentne wprowadzenie serii liczb, kombinacji liczb i liter, etykiet oraz obsługę dowolnej serii zdefiniowanej przez użytkownika. *SpeedFormat* umożliwia z kolei zastosowanie jednego z 25 zdefiniowanych wstępnie kolorowych formatów. Jest też opcja umożliwiająca tworzenie własnych formatów.

Nowa wersja Quattro Pro for DOS zawiera bogaty wybór sterowników do drukarek. Wyposażony jest też w złożone i wydajne narzędzia, ułatwiające bezproblemowe przejście do Quattro z arkusza Lotus 1-2-3 for DOS. Te narzędzia to m.in. menu zgodne z Lotusem, opcja *KeyReader*, uruchamiająca wykonywanie makroinstrukcji napisanych dla Lotusa 1-2-3 oraz możliwość odczytywania i zapisu plików w formacie Lotusa z zachowaniem stylów. Nabywcy Quattro Pro 5.5 for DOS (cenę ustalono na 59,95 USD) mogą korzystać przez sześć miesięcy z nieograniczonej pomocy technicznej serwisu Novella. (k)

Typologia kart płatniczych

dokończenie ze s. 13

Wymienione wyżej kierunki działań nie mogą się oczywiście wykluczać. Najważniejsze natomiast wydaje się skierowanie inwestycji banków polskich na drogę prowadzącą do zaoferowania naszemu społeczeństwu nowych, atrakcyjnych produktów bankowych, powodujących zadowolenie klienta, oraz docelowo wyraźny i określony wzrost poziomu technicznego oraz dochodu banku.

LITERATURA

[1] Card Word User Guide 1993, International Journal for the Plastic Card, Financial & Retail Services Industries – CARD WORLD PUBLICATIONS LTD, London, 1994

[2] Malicki A.: Raport z realizacji projektu karta płatnicza. Lublin 1994

[3] Svigals J.: Magnetic Stripes, Twenty Fifth Anniversary. ESCAT conference proceedings. Helsinki 1994.09.7–9.

Wydawnictwa Naukowo-Techniczne
00-048 Warszawa, Mazowiecka 2/4

POLECAJĄ! POLECAJĄ! POLECAJĄ! POLECAJĄ!

- Berglin S., Howes C., Jones S. - **Word 6 dla Windows** (przewodnik obrazkowy) -25,00 zł
- Booth D. - **Symphony 2** - 4,00 zł
- Botto F. - **Paradox** - 5,00 zł
- Dunn S., Bermant Ch., Berst J. - **Windows 3.1 - 101 wskazówek i trików** - 5,00 zł
- Ferbache D. - **Patologia wirusów** - 12,00 zł
- Harris W. - **Bazy danych nie tylko dla ludzi biznesu** - 12,00 zł
- Hertz J., Krogh A., Palmer R. G. - **Wstęp do teorii obliczeń neuronowych** - 12,00 zł
- Holland R. - **Testowanie i diagnostyka systemów mikrokomputerowych** - 9,00 zł
- Kernighan B. W., Ritchie D. M. - **Język ANSI C** -15,00 zł
- Klin M. Ch., Pöschel R., Rosenbaum K. - **Algebra stosowana dla matematyków i informatyków** - 6,00 zł
- Koblitz N. - **Wykład z teorii liczb i kryptografii** - 15,00 zł
- Kozielski S., Szczerbiński Z. - **Komputery równoległe. Architektura, elementy programowania** - 7,50 zł
- Kyle J. - **Zarządzanie pamięcią w Windows** -12,50 zł
- Leyland V. - **EDI. Elektroniczna wymiana dokumentacji** - 18,00 zł
- Nance B. - **Windows for Workgroups** - 21,00 zł
- Robling Denning D. E. - **Kryptografia i ochrona danych** - 13,50 zł
- Smith R. - **PostScript** - 25,00 zł
- Software W. - **SQL. Język relacyjnych baz danych** - 14,00 zł
- Southerton A., Perkins E. C., Jr. - **Słownik poleceń systemów Unix i X** - 43,00 zł
- Spence J. - **Excel 5** - 17,80 zł
- Stroustrup B. - **Język C++** - 34,00 zł
- Weiss Z., Gruźlewski T. - **Programowanie współbieżne i rozproszone w przykładach i zadaniach** - 9,00 zł
- Wood L. - **Poczta elektroniczna. Modemy i programy telekomunikacyjne** - 16,00 zł

Każdą z tych książek można nabyć za zaliczeniem pocztowym. Przy zamówieniu o wartości przekraczającej 25,00 zł koszty przesyłki pokrywa WNT, a dołączenie kuponu upoważnia do uzyskania 10% rabatu.

KUPON Informatyka 2/95

Nasz adres: Dział Handlowy WNT, skr. poczt. 359, 00-950 Warszawa, tel./fax 26-82-93

0121/95

SOFTARG '95

dokończenie z pierwszej strony okładki

mechanizm dzielenia słów dla 14, słowniki wyrazów bliskojęzycznych dla 11, tłumaczenie między 5. głównymi językami międzynarodowymi (tu brak polskiego, ale bez przesady: nasz język nie ma statusu języka kongresowego...). A możliwości edycyjne tego procesora nie tak wiele ustępują Wordowi. Ale uwaga: nie mogąc z braku czasu zbadać osobiście programu nie umiem nic powiedzieć o jakości jego spolszczenia – a robię to zastrzeżenie dlatego, że polszczyzna firmowej ulotki reklamowej jest tragiczna.

Calamus z kolei (rozpowszechniany przez firmę DTP z Chorzowa) był dotychczas powszechnie używany przez miłośników komputera Atari; miał również swą werję dla Macintosha. Teraz jest dostępny dla platformy Windows NT oraz dla Windows '95 – i jego możliwości są doprawdy ogromne. Nie ulega kwestii, że jest to jeden z najlepszych produktów w swej klasie; dostępne są wersje angielska i niemiecka. Na marginesie: myślę, że w przyszłości nie byłoby źle wprowadzić nowe konkurencje konkursowe: na nowość programową roku dla rynku polskiego oraz na najlepsze spolszczenie roku. Zwłaszcza ta druga wydaje mi się ważna: coraz więcej mamy na rynku spolszczeń – a niektóre z nich wołają po prostu o kij...

Co zaś do samego Softargu: chyba troszkę traci on swój charakter targów oprogramowania. Uczestnicy imprezy wystawiają taką masę sprzętu, że przytłacza ona wszystko inne – i troszkę szkoda. Warto także jakoś rozwiązać problem „wymiataczy", czyli tłumów młodych ludzi, wyraźnie przeszkadzających wystawcom w robieniu interesów. Choć – z drugiej strony – serce rośnie, gdy się patrzy jak wiele dzieciaków (i to płci obojga) entuzjazmuje się informatyką...

Bogdan MIŚ

WAŻNE!!!
Nie zapomnij o odnowieniu prenumeraty na rok 1996!

Książki nadesłane

WYDAWNICTWA NAUKOWO-TECHNICZNE

Anthony Lewis: Microsoft Excel 4.0. Z angielskiego przełożył Marian Łakomy. WNT, Warszawa 1993 r., wyd. 1, s. 136. ISBN 83-204-1652-3

Książka jest praktycznym wprowadzeniem do tego znakomitego arkusza obliczeniowego. Przeznaczona jest przede wszystkim dla początkujących użytkowników komputerów, którzy chcą szybko osiągnąć sprawność posługiwania się programem. Dobrze wybrane zadania zapoznają z podstawowymi funkcjami programu. Obszerny zestaw ćwiczeń służy do nabycia podstawowych umiejętności i umożliwia szybkie osiąganie postępu w posługiwaniu się arkuszem.

Książka może być używana jako podręcznik na kursach obsługi programu i do samodzielnej nauki. Umożliwia solidne opanowanie podstaw programu tak, aby czytelnik mógł tworzyć własne arkusze, mając pełne zaufanie do swych umiejętności.

Jim Kyle: Zarządzanie pamięcią w Windows™ 3.1. Z angielskiego przełożyła Elżbieta Sepko-Guzicka. WNT, Warszawa 1994 r., wyd. 1, s. 190. ISBN 83-204-1776-7

W teorii system Windows 3.1 jest dla użytkownika komputerów osobistych spełnieniem marzeń. Zawiera atrakcyjny i intuicyjnie łatwy w obsłudze graficzny interfejs użytkownika. Pozwala, by w tym samym czasie wiele programów było aktywnych. Stanowi oprogramowanie umożliwiające dostęp do dużo większej pamięci, dzięki czemu jest możliwe uruchamianie programów dużej mocy.

Książka informuje, jak zainstalować i skonfigurować system Windows, żeby jak najlepiej wykorzystać jego możliwości, wyjaśnia skróty dotyczące pamięci (EMS, XMS, HMA i UMB), podaje zasady umożliwiające skonfigurowanie planszy tak, aby zoptymalizować użycie pamięci. Znajdą się w niej wyjaśnienia, jak działają pliki CONFIG.SYS, AUTOEXEC.BAT, SYSTEM.INI i WIN.INI, jak radzić sobie z różnymi kłopotami systemowymi, co zrobić, by w systemie działały również inne programy do zarządzania pamięcią.

Książka, przeznaczona w zasadzie dla początkujących może stać się także pomocną dla doświadczonych użytkowników komputerów.

Zaprosili nas...

Amerykańska Agencja Rządowa ds. Rozwoju Handlu (*US Development Agency*) oraz **Stowarzyszenie Firm Telekomunikacyjnych** (*Telecommunications Industry Association*) na trzydniową konferencję dotyczącą globalnej sieci telekomunikacyjnej (GII). Omówiono telekomunikacyjną integrację Europy Środkowo-Wschodniej z siecią oplatającą kraje Europy Zachodniej oraz dalszą współpracę w dziedzinie infostrad, globalnej sieci komputerowej i komunikacji satelitarnej.

Wielkopolska Fundacja Tenisowa im. prof. Jana Fibaka na III konferencję prasową zorganizowaną z okazji zawodowego turnieju tenisowego ATP Tour Challenger COMPAQ POLISH OPEN '95, sponsorowanego przez firmę Compaq Computer.

Prezes Urzędu Zamówień Publicznych, Pan Józef Żak, na konferencję prasową podsumowującą półroczne obowiązywanie ustawy o zamówieniach publicznych. Omówiono także nowelizację ustawy, uchwaloną przez Sejm w lipcu br.

Microsoft Sp. z o.o. na konferencję prasową z powodu światowej premiery nowego systemu operacyjnego do komputerów osobistych Microsoft Windows '95 oraz na przyjęcie z tej okazji do Muzeum X. Dunikowskiego.

MSP/TH 'System oraz **IBM Polska** na konferencję prasową i uroczyste przyjęcie organizowane w związku z podpisaniem przez te firmy umowy dystrybucyjnej.

Digital Equipment i **Microsoft** na wspólną konferencję, w trakcie której omówiono współpracę obu firm i przedstawiono informację o nowych produktach.

Apple i **A.C.I., Sad** oraz **4 Dream** na wspólną konferencję, na której zaprezentowano możliwości międzyplatformowe narzędzi programowych i sprzętowych, łatwość ich użycia na wszystkich poziomach użytkowania oraz integrację z istniejącymi systemami.

BPS do odwiedzenia swojego stoiska podczas wystawy towarzyszącej XI Krajowemu Sympozjum Telekomunikacji w Bydgoszczy. Szczególną uwagę przyciągał najnowszy produkt BPS – system teletransmisji WATSON, umożliwiający uzyskanie dodatkowych 30. numerów abonenckich oraz możliwość jednoczesnego przepływu danych, obrazu i dźwięku z szybkością 2 Mb/s.

Oficyna Informatyczna i **Fundacja Kultury Informatycznej** na spotkanie o tematyce – CorelDraw 5.0 a 6.0 – możliwości zastosowań. (t)

Perfect-Ekspert AITECH

20 LAT DOŚWIADCZEŃ WDROŻENIOWYCH

Sieciowy i zintegrowany system informatyczny, głównie dla dużych firm produkcyjnych i handlowych. Przystosowany do elektronicznej wymiany dokumentów (EDI - zamówienia, faktury, WZ, itp.), wielowalutowości, denominacji złotego oraz rachunkowości zarządczej. Wyposażony w system ochrony danych generowanych wewnątrz i na zewnątrz przedsiębiorstwa. Współpracujący z systemem ekspertowym ISAF (Inteligentny System Analiz Finansowych) wspomagający **podejmowanie decyzji finansowych na średnich i wyższych szczeblach zarządzania.**

MAXIVELL

System Informacji Kierownictwa	Inteligentny System Analiz Finansowych	System Finansowo-Kosztowy	System Wyposażenie
System Obrotu Towarowego i Fakturowania	System Obrotu Materiałami i Ewidencji Zakupu	Rozliczenie Zakupu Materiałów	System Środki Trwałe
System Kadry - Płace	System Analiz Marketingowych	System Technicznego Przygotowania Planowania i Rozliczania Produkcji	EDI TMW Ultra - Perfect

Oferujemy kompleksowe usługi w zakresie:
- doradztwa organizacyjnego, merytorycznego, technicznego, consulting,
- organizacji, projektowania i wykonawstwa innych systemów informatycznych,
- wdrażania, konserwacji i modernizacji systemów informatycznych,
- projektowania, wykonawstwa dowolnego typu sieci komputerowych, z uwzględnieniem już istniejących rozwiązań,
- dostawa sprzętu komputerowego i sieciowego,
- szkoleń użytkowników w zakresie operatora i administratora systemu AS/400, jak również obsługi pakietów programowych PC Support/400 lub Client Access/400 przeznaczonych do współpracy z PC.

JUNISOFTEX

44-100 Gliwice, ul. Kościuszki 1 c, box 507
Tel./fax (0-32) 31-51-31, 31-57-00
38-80-52, 38-80-54

JUNISOFTEX

1/95

KARTA PKO VISA CLASSIC TWOJĄ DEWIZĄ

Na całym świecie zastąpi Ci gotówkę, czek i kartę telefoniczną.

Przy użyciu karty **PKO VISA Classic** zarówno w Polsce jak i za granicą możesz płacić swoimi pieniędzmi zgromadzonymi w PKO BP bez konieczności posługiwania się gotówką lub czekiem.

Karta **PKO VISA Classic** umożliwia pobieranie gotówki w bankomatach i kasach banków na całym świecie.

Chcąc płacić kartą **PKO VISA Classic** za granicami Polski nie trzeba blokować środków na rachunku dewizowym.

Bank raz w miesiącu rozlicza wszystkie transakcje dokonane przy użyciu karty, obciążając w kolejności określonej przez klienta banku jego rachunki w PKO BP.

Kartę **PKO VISA Classic** można uzyskać już po roku posiadania konta osobistego w Powszechnej Kasie Oszczędności.

PKO

0127/95